ARMY RESEARCH LABORATORY

LADAR – Camera Registration Using Object Pose and Camera Calibration Algorithms

By William F. Oberle and Gary A. Haas

ARL-TR-3147 **April 2004**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

DESTRUCTION NOTICE—Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-3147 **April 2004**

LADAR – Camera Registration Using Object Pose and Camera Calibration Algorithms

William F. Oberle and Gary A. Haas
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
April 2004	Final	January 2004 to March 2004

4. TITLE AND SUBTITLE

LADAR – Camera Registration Using Object Pose and Camera Calibration Algorithms

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

William F. Oberle; Gary A. Haas (both of ARL)

5d. PROJECT NUMBER
622618AH03

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
Weapons and Materials Research Directorate
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-3147

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

To be effective, combining or fusing data from a variety of sensors requires all the data to be placed within the context of a common coordinate system. The process of determining the relation between the various sensor coordinate systems is termed "registration". This work investigates the registration between a LADAR sensor and an imagery camera. Both object pose and camera calibration algorithms are explored. The two methods providing the best results are an object pose method, POSIT, by DeMenthon and Davis, University of Maryland, College Park, and a camera calibration method by Bouguet.

15. SUBJECT TERMS

camera calibration; LADAR; object pose; sensor registration

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			William F. Oberle
Unclassified	Unclassified	Unclassified	UL	50	19b. TELEPHONE NUMBER (*Include area code*) 410-278-4362

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgments

The authors would like to thank Thomas Haug of the U.S. Army Research Laboratory (ARL) for his time and effort in reviewing the report. His comments and suggestions proved valuable in the preparation of the final version of the report. Furthermore, we would like to thank Dr. Daniel DeMenthon of the University of Maryland, College Park, Maryland, for providing the POSIT code and the many discussions concerning the use of the code.

This work was the outgrowth of research performed under Cooperative Research and Development Agreement No. 0312-A-C799 between the University of Maryland and ARL.

INTENTIONALLY LEFT BLANK.

1. Introduction

To accomplish autonomous mobility, an unmanned ground vehicle (UGV) relies on data supplied by a variety of sensors as it navigates its way through unfamiliar terrain. These mobility sensors can basically be classified into two categories: those that provide data concerning the location and orientation of the UGV relative to the world (navigation) and those that detect and provide data concerning the location of objects within the immediate environment of the UGV (obstacle detection). Sensors in the former category include inertial navigation systems and global positioning systems, while laser radar (LADAR) and imagery cameras (monocular or stereo) are examples of sensors in the latter category. Although current UGV architectures integrate or fuse[1] navigation data, it does not appear that the same can be said for obstacle detection data. Yet, only through the integration or fusion of all available information can the data provided by the obstacle detection sensors be fully exploited.

Essential to the data fusion procedure is the capability of expressing all data to be fused within the same reference frame. For obstacle detection sensors, the common reference frame of interest is preferably the same coordinate system used by the navigation sensors but more likely, the coordinate system associated with one of the sensors. Each sensor records data relative to its own individual coordinate system. Assuming mutually orthogonal coordinate axes with the same relative orientation (i.e., right-handed or left-handed coordinate system), at any instance in time, there exists a rigid body transformation (RBT) between the coordinate system of any pair of sensors. This RBT is parameterized by translations along and rotations about the three coordinate axes expressed as a rotation matrix and translation vector. Determining the RBT between sensors is referred to as "sensor registration."

Of interest in this report is sensor registration between a LADAR sensor and a calibrated[2] imagery camera. In this case, the sensor registration problem is equivalent to the object pose problem, i.e., computation of the rotation matrix and translation vector between a three-dimensional (3-D) object(s) and a camera, given a number of correspondences (matches) between 3-D object(s), reference points, and their images (DeMenthon and Davis, 1995).[3] Likewise, the LADAR-camera registration problem is conceptually similar to camera calibration. However, camera calibration is more general in that the camera calibration procedure estimates the intrinsic camera parameters and the registration RBT at the same time.

[1] Hall and Llinas define data fusion by "data fusion techniques combine data from multiple sensors and related information from associated databases to achieve improved accuracy and more specific inferences that could be achieved by the use of single sensor alone" (Wald, 1999).

[2] Calibrated camera refers to a camera for which the intrinsic parameters are known. The intrinsic camera parameters include the image center (pixels), focal length (millimeters), physical dimensions of the image pixels (millimeters), and the angle between x-axis and y-axis of the pixels. See Trucco and Verri (1998) for additional details.

[3] Some authors require that the entire geometry of the 3-D object be known for the object pose problem. In this report, a less restrictive viewpoint is taken (i.e., only a small number of points on the 3-D object must be specified).

LADAR sensors provide 3-D coordinates (relative to the LADAR coordinate system) of objects within their field of view (FOV), whereas a camera provides a projective two-dimensional (2-D) image (i.e., measured intensity of reflected light) of objects in its FOV. Although distance measurements are not given by the camera data, the geometry of the camera does provide information relative to lines of sight (LOS) of objects relative to the camera coordinate system (see figure 1). In the figure, let $\boldsymbol{P}$ represent a world (LADAR) point in the FOV of the camera and $\boldsymbol{P}_1$ the image of $\boldsymbol{P}$. If the camera is calibrated, the 3-D coordinates of $\boldsymbol{P}_1$ in the camera coordinate system are known. Thus, the ray $\boldsymbol{OP}_1$ defines the LOS for the world (LADAR) point $\boldsymbol{P}$ in the camera coordinate system. In the setting of the object pose problem, the LADAR-camera registration problem can be posed as determining the RBT that maps the 3-D LADAR points into the camera coordinate system so that the mapped points lie on the appropriate LOS determined by the camera image points. This is also equivalent to projecting the 3-D LADAR onto the camera image in such a manner that the projected points coincide with the matched image points. Generally, one of two methods to solve the object pose problem is used. In the first method, optimization techniques are used to determine the RBT that minimizes the difference between the pixel coordinates of the projected 3-D points and their corresponding image points. The second method employs a two-step approach. First, 3-D coordinates of the known object points are determined in the camera coordinate system via the law of cosines.[4] This is followed in the second step by the determination of the RBT between the two sets of 3-D points (i.e., 3-D object points expressed in the world and camera coordinate systems). A more detailed description of the object pose problem is presented in Quan and Lan (1999) and references therein. For n matching points, Fischler and Bolles (1981) refer to this as the perspective-n-point problem (PnP problem). The camera calibration problem is described in almost all books on computer vision (e.g., Faugeras, 1993, or Trucco and Verri, 1998) and has been the focus of intense research in the vision community. Generally, methods to solve the camera calibration problem use the matched points and a camera model (e.g., pin-hole perspective camera) to generate a system of linear equations with the unknowns being the components of the RBT rotation matrix and translation vector, and intrinsic parameters.

The remainder of the report is organized as follows. In section 2, the criteria for selecting a method for solving the LADAR-camera registration (object pose/camera calibration) problem are discussed, and candidate methods are identified. Section 3 provides details of the synthetic and experimental data used in evaluating candidate methods. Procedures used in evaluating the

4Consider two 3-D points, p1 and p2, on the object. Since the 3-D coordinates of p1 and p2 are known in the LADAR coordinate system, the distance between p1 and p2 can be determined. This distance is the same in both the camera and LADAR coordinate systems. Consider the triangle formed by p1, p2, and the camera center, $\boldsymbol{O}$. Since the LOS to p1 and p2 can be determined, the angle $\angle$p1$\boldsymbol{O}$p2 is known. Thus, the law of cosines can be used to write an equation expressing the distances $\boldsymbol{O}$p1 and $\boldsymbol{O}$p2 (in the camera coordinate system) as a function of the distance p1p2 and the angle $\angle$p1$\boldsymbol{O}$p2. If three points are considered, then three quadratic equations in the unknowns $\boldsymbol{O}$p1, $\boldsymbol{O}$p2, and $\boldsymbol{O}$p3 can be solved simultaneously to obtain values for $\boldsymbol{O}$p1, $\boldsymbol{O}$p2, and $\boldsymbol{O}$p3. It is then a simple calculation to determine the 3-D coordinates of p1, p2, and p3 in the camera coordinate system. The solution of the three simultaneous quadratic equations reduces to finding the roots of a single fourth order polynomial.

candidate methods are discussed in section 4. Evaluation results are provided in section 5. Finally, section 6 provides a discussion of the results and conclusions.

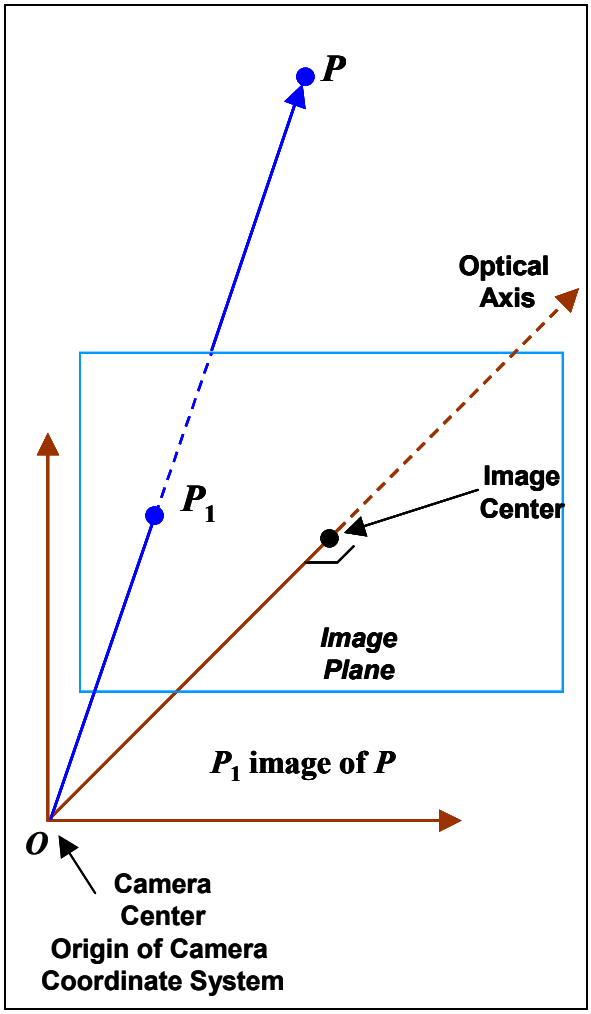

Figure 1. Schematic of camera coordinate system and LOS to world point P.

2. Criteria and Candidate Methods

As stated in the introduction, the intended use of the selected method is to determine the registration between a LADAR sensor and camera. These sensors will be mounted on an UGV traveling primarily over outdoor terrain. Although it is envisioned that the LADAR sensor and camera (most likely a stereo camera pair) will be rigidly mounted relative to each other within the "stereo rig," periodic registration of the devices is likely to be necessary because of the vibration of the UGV as it travels over potentially rough terrain. Considering this operational

domain, it would be desirable to have the candidate methods meet the following criteria. The method should

1. be capable of handling real-world outdoor data typical of the UGV environment.
2. be capable of working with sets of matched or control points that have no specific geometric construct.
3. be capable of handling a wide range in the number of matched points.
4. be capable of determining the registration with run times of less than several minutes.
5. not require input data beyond the matched points and camera calibration information.
6. provide a result without user intervention.

The object pose and camera calibration problems have been studied by a large number of researchers (e.g., DeMenthon and Davis, 1995; Haralick, Lee, Ottenberg, and Nolle, 1991; Fischler and Bolles, 1981; Quan and Lan, 1999; Tsai, 1986 and 1987; Trucco and Verri, 1998, Faugeras, 1993; Willson, 1995; Baeten, 2003; Bouguet, 2003). None of the methods investigated for solving the object pose or camera calibration problems provided results for UGV environment-like data. In addition, results reported in the literature with these methods generally involved structured scenes, e.g., calibration posters. Thus, the first two criteria cannot be addressed on the basis of the literature review, nor will these two criteria be addressed in this report since the experimental data would also be considered a structured scene.

For the object pose problem, the majority of methods ultimately require that the roots to a number of fourth order polynomials be determined (footnote 4, Quan and Lan, 1999). Both iterative and analytic methods have been applied to solving the required polynomials. However, as Quan and Lan (1999) state, "Iterative methods suffer from the problems of initialization and convergence, while algebraic methods applied to subsets suffer from poor noise filtering and the difficulty of selecting the common root from the noisy data." To overcome these shortcomings, different researchers have imposed additional constraints on the problem to assist in determining the solution consistent with the input data. Fischler and Bolles (1981) introduced the random sample consensus (RANSAC) paradigm to minimize the impact of data outliers but also introduced the requirement to specify values for three unspecified parameters. Even after the input of outliers is purged, a separate method has to be applied to calculate the results. Others have restricted the image features associated with the matched points, generally lines or rectangles instead of points (Horaud, Conio, and Leboulleux, 1989; Abidi and Chandra, 1995). Still others have combined alternate image features with probabilistic approaches (Hanek, Navab, and Appel, 1999). However, the necessity to determine the roots of the fourth order polynomial remains. In addition, restricting image features violates the second criterion. Criterion number 3 eliminates those methods addressing the PnP problem for small values of n. Methods using a Newton-Raphson or similar approach to approximate the roots of the fourth

order polynomial generally require a user-supplied initial estimate and do not satisfy the fifth criterion.

Solutions to the camera calibration problem generally involve solving a system of linear equations. In general, the methods for solving systems of linear equations are consistent with the criteria listed previously. Thus, it would appear that camera calibration approaches offer the better possibility to address the LADAR-camera registration. Unfortunately, as Faugeras (1993, page 66) states, "But the fundamental instability of the (camera calibration) problem, ..., has delayed the availability of a robust and flexible technique, ..." This leads to most camera calibration methods being ill posed mathematically and highly susceptible to error because of noisy input data. To mitigate input noise, special surveyed calibration posters are often employed which is contrary to the second criterion.

Cobzas, Zhang, and Jagersand (2002) addressed the registering of 3-D LADAR data to 2-D image data. Their conclusion was that point-based calibration methods performed best to recover a global registration between the sensors. Based on their observation and the previous discussion, both object pose and camera calibration methods operating directly on the matched points are selected as candidate methods for the LADAR-camera registration.

The object pose method is by DeMenthon and Davis (1995) and termed "POSIT" (pose from orthography and scaling with iterations). Although this method does not guarantee convergence to a solution, testing performed with POSIT produced solutions in more than 99% of the test cases. Two camera calibration methods are chosen. The first is by Tsai[5] (1986, 1987) and the second by Bouguet (2003). All three methods appear to have reasonable run times, best meet the criteria, and are highly recommended in the literature. Code for the POSIT calculations was provided by DeMenthon (January 2004). The Tsai method calculations were performed with Baeten's (2003) MATLAB[6] version of Willson's (1995) C implementation of Tsai's method, both of which are available on line. The *Camera Calibration Toolbox for MATLAB* (available on line) (Bouguet, 2003) is used in the calculations based on Bouguet's method.

3. Test Data

The test data consist of a combination of experimentally obtained and synthetically generated data. Experimental data used in this report were collected at the campus of the National Institute of Standards (NIST) in May 2003. A Riegl Z210 Laser Measurement Systems (LMS) (see references) LADAR sensor provided by NIST and a stereo camera pair (Sony DXC[7] 9000

[5] Non-coplanar calibration with full optimization option used.

[6] MATLAB® is a registered trademark of The MathWorks.

[7] Not an acronym

cameras) provided by the U.S. Army Research Laboratory (ARL) were used. A picture of the LADAR sensor and cameras mounted on the HMMWV (high mobility multipurpose wheeled vehicle) used for the data collection is shown in figure 2. Note that the LADAR sensor is mounted above, behind, and to the right of the stereo camera pair (relative to the camera coordinate system). A schematic of the LADAR and camera coordinate systems is shown in figure 3. The sensor coordinate systems are nearly aligned toward the scene, but the orientation of the axes is quite different. As will be seen in sections 5 and 6, this orientation of the two coordinate systems presents difficulties for the two camera calibration methods. NIST and ARL personnel jointly collected the data.[8] An example of the imagery data recording the intensity of the reflected laser pulses recorded by the Riegl LADAR sensor is shown in figure 4. Image resolution is 536 by 616 (column by row) pixels and the image is rotated counter-clockwise by 90 degrees because of the manner in which the LADAR data are collected. For each pixel in the intensity image, the Riegl LADAR software calculates the distance estimates as part of its standard output. The left and right images from the stereo cameras are shown in figure 5. Images are 640 by 480 (columns by row) pixels.

Figure 2. LADAR and cameras mounted on HMMWV used for data collection.

In order to perform the registration calculations, the authors had to match the 3-D coordinates for points from the LADAR to 2-D pixel locations in the left and right camera images. The matching process is performed manually. For the purposes of identifying the matched points, the LADAR

[8] POCs: G. Haas (ARL/WMRD); P. Gillespie (ARL/SEDD); M. Shneier (NIST); H. Scott (NIST).

intensity image is used in combination with the camera image to select corresponding pixels in the three images.

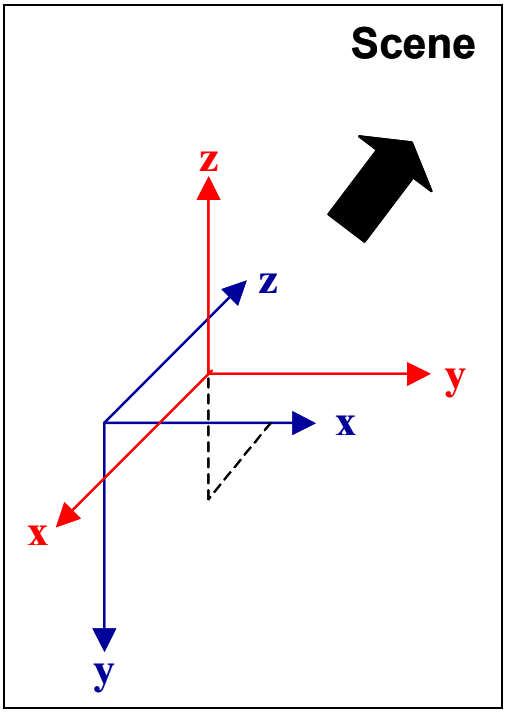

Figure 3. Schematic of LADAR (red) and camera (blue) coordinate systems. (LADAR is behind, above, and to the right of the camera.)

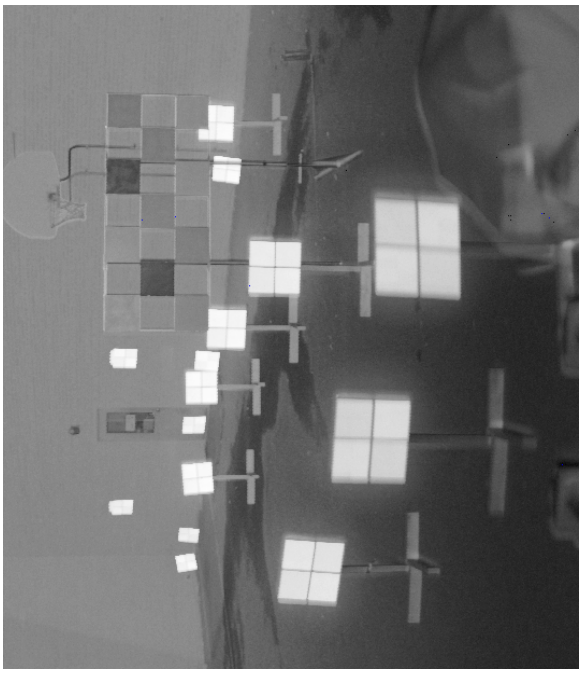

Figure 4. Intensity image recorded by Riegl LADAR.

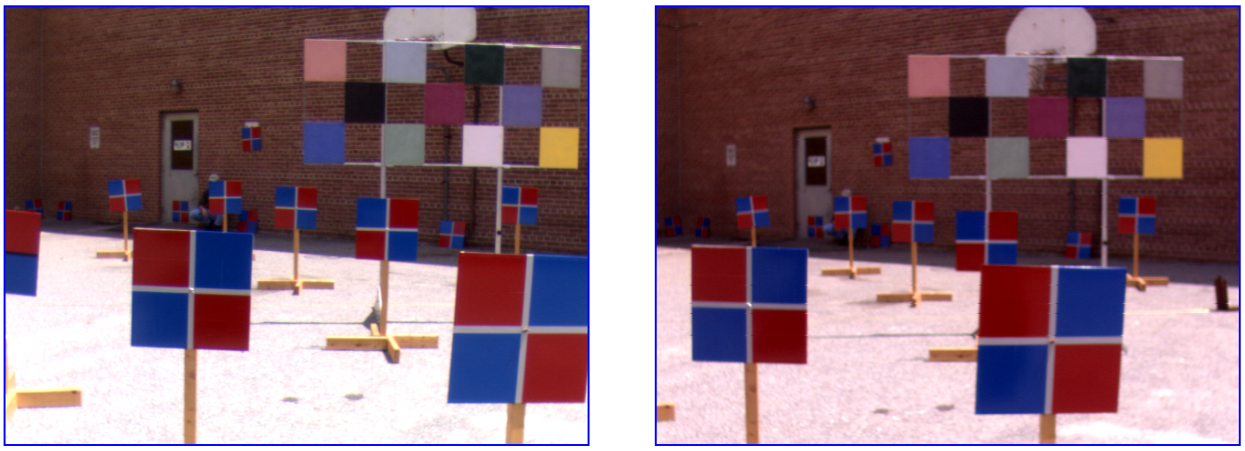

Figure 5. Left and right images from stereo camera pair.

The basic steps in the manual matching are

1. Identify a feature common to all three images (LADAR intensity, left and right cameras).
2. Determine by inspection the pixel location of the feature in each image.
3. Use the Riegl output to match the feature pixel location in the LADAR intensity image with its 3-D coordinates.

With this approach, 36 corresponding points were identified (see figures 6 through 8). However, manually matching points with the rotated LADAR intensity image in figure 4 with points in an upright camera image in figure 5 proved to be erroneous and tedious. Thus, the decision was made to rotate the LADAR intensity image. Rotating the image is easily accomplished with any image processing software. In the rotated image, pixel coordinates are mapped to new values to be consistent with the new orientation of the image and standard image notation. Figure 9 shows a schematic of the situation. The mapping of the pixel locations would be transparent except for the fact that the 3-D locations are associated with the pixels in the coordinates of the original intensity image, not the rotated intensity image. Thus, it is necessary to format the 3-D LADAR data to account for the pixel location change because of the rotation of the image.

Figure 10 shows the header information and the first two lines of the original 3-D LADAR data file (besides the header information, there are as many as 536 * 616 = 330176 lines of 3-D position data). Each line of data consists of 11 entries identified in the header line VARIABLES. The first three entries are the 3-D coordinates (meters) of the object in the pixel. Entry number 4 is the distance to the object computed with the standard Euclidean distance formula. This is followed by six entries not used in this report, which will not be discussed. The final entry is the *measure id* (D) that relates the data to a pixel location as defined by lines 3 through 5 of the header information.

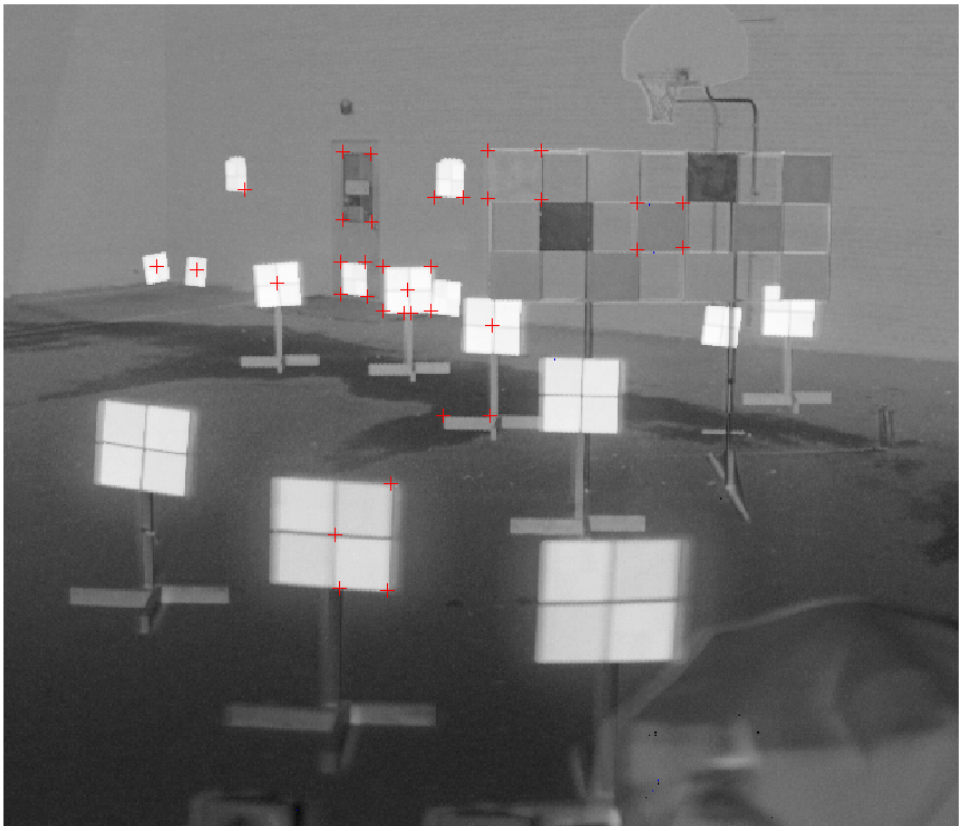

Figure 6. Matched points (+) in LADAR intensity image.

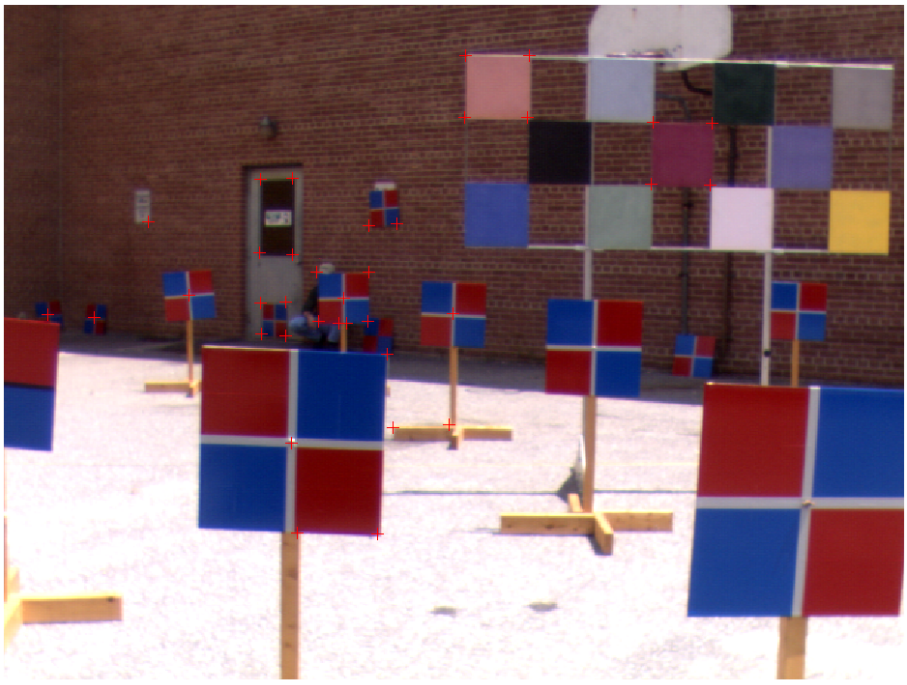

Figure 7. Matched points (+) for left stereo image.

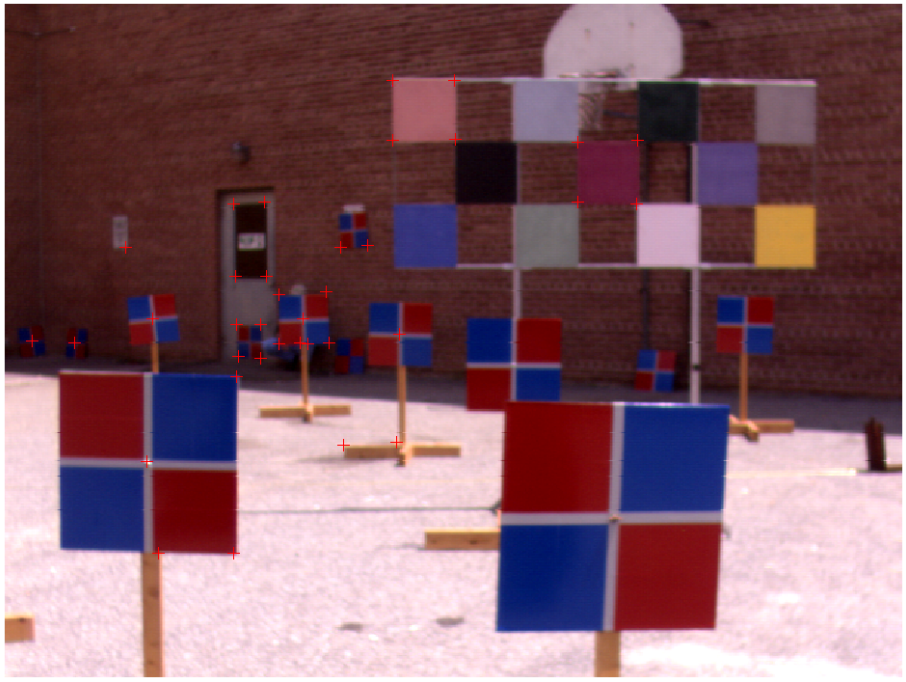

Figure 8. Matched points (+) for right stereo image.

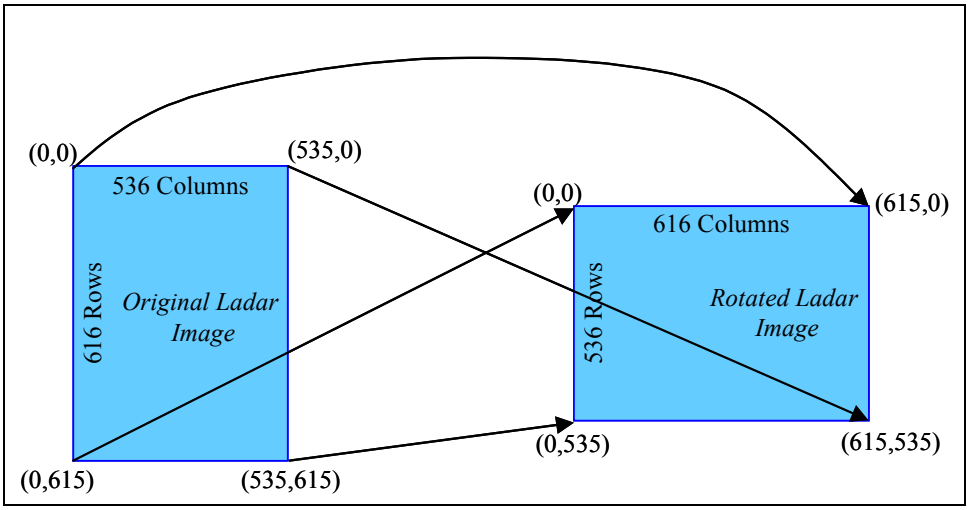

Figure 9. Schematic of original and rotated LADAR intensity image.

```
# ASCII Export File
# Automatically generated by - 3D-RiSCAN
# 3D data size - meas  / line:  536
#              lines / image: 616
# meas id = meas# + line# * (meas/line)
# VARIABLES: X[m] Y[m] Z[m] R[m]_COR I cl_8_R cl_8_G cl_8_B P[deg]_COR
A[deg]_COR D
# FORMAT: %8.3f %8.3f %8.3f %8.3f %4.0f %4.0f %4.0f %4.0f %9.3f %9.3f %9.0f
#
 -11.903   3.789   2.608  12.761  140  255  255  255   78.209  162.344       0
 -11.921   3.794   2.595  12.777  143  255  255  255   78.281  162.344       1
```

Figure 10. Header information and first two lines of the Riegl LADAR 3-D data output.

If $(\mathbf{X_O},\mathbf{Y_O})$ represents the pixel coordinates in the original intensity image, $(\mathbf{X_R},\mathbf{Y_R})$ the corresponding pixel coordinates in the rotated image, and ***index*** the *measure id* for a line of data, then the following defines the mapping from the data in the original 3-D LADAR file to 3-D locations relative to the rotated LADAR intensity image. Line 5 of the header information in figure 10 defines the mapping from the *measure id* to pixels in the original intensity image. First,

$$\mathbf{X_O = meas\#} \quad \text{and} \quad \mathbf{Y_O = line\#},$$

which implies using line 5,

$$\boldsymbol{index} = \mathbf{X_O + 536 * Y_O}. \tag{1}$$

Now the relation between ***index*** and the pixel location in the original intensity image is defined by the standard approach used to store a 2-D array in a one-dimensional (1-D) array. Thus,

$$\mathbf{X_O} = \boldsymbol{index} \ \mathbf{mod\ 536}, \tag{2}$$

and substituting equation 2 into equation 1 and solving for $\mathbf{Y_o}$, yields

$$\mathbf{Y_O = (}\boldsymbol{index}\mathbf{ - X_O) / 536}.$$

Once the relation between the ***index*** and pixel coordinates in the original intensity image is computed, the mapping to the pixel coordinates in the rotated intensity image is given by

$$\mathbf{X_R = 615 - Y_O} \quad \text{and,} \quad \mathbf{Y_R = X_O}.$$

Appendix A provides a listing of the C++ program used to format the original LADAR data to correspond to the rotated intensity image. A complete listing of the matched points and the 3-D locations is provided in appendix B. A listing of the C++ code used to read the data for the rotated intensity image (created with the program in appendix A) and determine the 36 3-D locations is provided in appendix C.

The candidate methods either assume calibrated cameras (POSIT) or require certain intrinsic parameters for each camera. The required values are provided in table 1. Each candidate method requires different intrinsic parameter input. The superscript indicates that if the parameter is required for a given method, P – POSIT, T – Tsai, B - Bouguet.

Table 1. Intrinsic parameters for the cameras used in the study.

Intrinsic Parameter	Explanation	Left Camera	Right Camera
Focal Length (pixel)^{P,B}	Camera focal length/physical size of picture element	865	852
Center – X^{P,T,B} (pixel)	Horizontal pixel of intersection of optical axis and image plane	336	348
Center – Y^{P,T,B} (pixel)	Vertical pixel of intersection of optical axis and image plane	248	266
Ncx^{T}	Number of elements in a single row charge-coupled device (CCD) array	640	640
Nfx^{T}	Number of pixels in a single row CCD array	640	640
dx^{T} (millimeter)	Horizontal distance between adjacent cells CCD array	0.01	0.01
dy^{T} (millimeter)	Vertical distance between adjacent cells CCD array	0.01	0.01
sx^{T}	Scale factor	1	1
Kc^{B}	Distortion coefficients	All 0	All 0
alpha_c^{B} (radians)	Skew coefficient	0	0

4. Evaluation Procedure

As mentioned in the introduction, the goal of the LADAR-camera registration is to determine the RBT that maps 3-D coordinates expressed in the coordinate system of the LADAR to 3-D coordinates in the coordinate system associated with the camera. This relationship can be expressed in one of two formats. The two formats differ, depending on which coordinate system is used to express the translation. If $\boldsymbol{R}$ is the rigid body rotation, $\boldsymbol{p}_C$ a point represented in the camera coordinate system, and $\boldsymbol{p}_L$ the same point in the LADAR coordinate system, then both of the following expressions can be use define the RBT between the LADAR and camera:

$$\boldsymbol{p}_C = \boldsymbol{R}\,\boldsymbol{p}_L + \boldsymbol{T}_C \qquad \text{or} \qquad \boldsymbol{p}_C = \boldsymbol{R}(\,\mathbf{p}_L - \boldsymbol{T}_L) \tag{3}$$

In the expression on the left, the translation is relative to the camera coordinate system, while the translation is relative to the LADAR coordinate system for the expression on the right. $\boldsymbol{T_C}$ and $\boldsymbol{T_L}$ are related by the relations

$$T_C = -RT_L \qquad \textbf{\textit{or}} \qquad T_L = -R^{-1}T_C.$$

Both formats are used in the report; the format given to the left in 3 is referred to as the rigid body A (RBA) and the format to the right as rigid body B (RBB).

4.1 Synthetic Data

Since no ground truth (i.e., the RBT between the LADAR and either camera image) is available, a set of synthetic image data is created with the 36 LADAR 3-D points from the set of matched points together with an arbitrary RBT. The RBT selected in RBA format is

$$\boldsymbol{R_S} = \begin{pmatrix} 0.08749666119173 & 0.99566917244726 & 0.03142026923014 \\ -0.04587594117370 & 0.03553521160292 & -0.99831490360395 \\ -0.99510789982882 & 0.08590778646042 & 0.04878647275366 \end{pmatrix}$$

and

$$\boldsymbol{T_S} = \begin{pmatrix} -0.33448540934011 & 0.89782925664977 & 15.22788837258931 \end{pmatrix}.$$

A MATLAB (2001) function to map the 3-D LADAR coordinates to camera pixel coordinates for a given RBA RBT and a camera with a given center and focal length (intrinsic parameters) is given in appendix D. The intrinsic parameters used in the calculation are those for the left-hand camera (see table 1). Details about transforming 3-D points in the camera coordinate system to pixel units are given in Trucco and Verri (1998). The synthetic pixel data are provided in appendix E. Note that the synthetic pixel data are given to subpixel accuracy. Thus, if the 3-D LADAR points and the matching synthetic pixel data are used as input, then the RBT defined by $\boldsymbol{R_S}$ and $\boldsymbol{T_S}$ becomes the ground truth for the calculation.

With the 3-D LADAR-synthetic pixel data as input, three different calculations will be performed for each candidate method:

1. Estimate the LADAR-camera registration, i.e., RBT, for each candidate method with all 36 3-D LADAR-synthetic pixel data combinations as input. Compare the results to the RBT defined by $\boldsymbol{R_S}$ and $\boldsymbol{T_S}$ in order to assess the method's capability to determine the LADAR-camera registration during ideal conditions (i.e., noise-free input data). See appendix F for MATLAB function to compare RBTs. Details about the comparison methodology are given in section 4.3.

2. In order to access the each candidate method's stability with different number of input points, use randomly selected subsets of the 36 3-D LADAR-synthetic pixel data combinations as input for each candidate method and estimate the LADAR-camera registration. Use 10 subsets each of 4 through 35 points. Compare the results to the RBT defined by $\boldsymbol{R_S}$ and $\boldsymbol{T_S}$. See appendix G for a MATLAB script file to randomly select subsets from the matched points and perform the calculation and comparison.

3. In order to access each candidate method's capabilities to deal with noisy input data, round the synthetic pixel data to the nearest integer pixel and estimate the LADAR-camera registration for each candidate method with all 36 3-D LADAR-rounded synthetic pixel data combinations. Compare the results to the RBT defined by $\boldsymbol{R_S}$ and $\boldsymbol{T_S}$.

4.2 Experimental Data

The experimental data consist of the 36 matched 3-D LADAR points and pixel locations in the left and right camera images (see appendix B). Although there is no ground truth (i.e., registration) relating the LADAR and cameras, results from the candidate methods with the

experimental data as input will be analyzed in two ways to assess the "goodness" of the calculated registrations (LADAR-left camera and LADAR-right camera).

First, an approximate registration between the left and right cameras is available from earlier work (Oberle, 2004). The LADAR-left camera and LADAR-right camera registrations calculated by each candidate method with the experimental data can be combined to provide an estimate of the registration between the left and right cameras. This estimate can be compared to the approximate registration in Oberle (2004). The format of the registration in Oberle (2004) is RBB with 3-D points in the coordinate system of the left camera being mapped to 3-D points in the right camera. This registration is defined by

$$\boldsymbol{R}_{\boldsymbol{LR}} = \begin{pmatrix} 0.9998008 & 0.0164890 & -0.0104682 \\ -0.0163086 & 0.9998658 & -0.0038292 \\ 0.0104173 & 0.0039948 & 0.9999360 \end{pmatrix}$$

and

$$\boldsymbol{T}_{\boldsymbol{LR}} = \begin{pmatrix} 334.761 & -1.676 & -3.927 \end{pmatrix}.$$

The units for the translation $\boldsymbol{T}_{\boldsymbol{LR}}$ are in millimeters. Since the units for the LADAR data are meters (see appendix B), any translation vector determined by a candidate method will have to be converted to compatible units with $\boldsymbol{T}_{\boldsymbol{LR}}$ before comparison.

The second approach to assess the goodness of the LADAR-left camera and LADAR-right camera registrations calculated by each candidate method is to use the registration to project the 3-D LADAR points onto the left (right) camera image with a pinhole projective camera model and compare the projected points to the actual pixel coordinates in the experimental data. However, care should be exercised when one is interpreting the results for this calculation. Although there is a unique registration between the LADAR sensor and camera, there may be alternate RBTs that result in "acceptable" projections of the 3-D LADAR points onto the image plane. An application such as integrating color information from a camera image with point location information observed by a LADAR sensor (common in UGV terrain classification) might work adequately with such a registration, while another application such as fusing 3-D observations from LADAR with 3-D observations from stereoscopic vision would require a more correct RBT.

Therefore, the procedure to evaluate the candidate methods with the experimental data involves five steps.

1. Use the 36 matched experimental points for the LADAR and left camera to determine the RBT between the LADAR and left camera.
2. Repeat step 1 for the right camera with the experimental points for the LADAR and right camera.

3. Use the results of steps 1 and 2 to determine the RBT between the left and right cameras.
4. Compare the result of step 3 to $\boldsymbol{R_{LR}}$ and $\boldsymbol{T_{LR}}$.
5. Project the LADAR points to pixel coordinates with the RBTs determined in steps 1 and 2. Use the intrinsic parameters in table 1 and the code in appendix D to perform the projection. Determine the mean and standard deviation between the projected pixel values and the actual pixel values for the left and right camera images. The MATLAB script file to perform this operation is given in appendix H.

4.3 Comparing Rigid Body Transformations

An RBT is defined by a rotation and translation. Thus, to compare two RBTs, metrics to compare both the rotations and translations must be defined. In this report, the metric for comparing rotations is

$$\text{Error}(\boldsymbol{R_1}, \boldsymbol{R_2}) = \text{frob}\left(\frac{\boldsymbol{R_1}}{\text{frob}(\boldsymbol{R_1})} - \frac{\boldsymbol{R_2}}{\text{frob}(\boldsymbol{R_2})} \right),$$

$\boldsymbol{R_1}$ and $\boldsymbol{R_2}$ are two rotation matrices, and frob is the Frobenius norm defined for the n-by-m matrix $[\boldsymbol{A_{ij}}]$ as

$$\text{frob}\left(\left[\boldsymbol{A_{ij}} \right] \right) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \left(\boldsymbol{A_{ij}} \right)^2} \ .$$

Manning (2003) states that if Error($\boldsymbol{R_1}$,$\boldsymbol{R_2}$) < 0.001, excellent 3-D reconstructions result. This value will be used as the figure of merit in comparing the rotations.

Translation vectors are defined by their magnitude and direction. The percentage difference in the translation vector norms will be used to compare the magnitudes. Directions will be compared via the scalar vector product (dot product) to determine the angle between the two translation vector directions. No figure of merit for these values was found in the literature.

Thus, three measures will be used in comparing the RBTs.

5. Evaluation Results

5.1 Synthetic Data Results

5.1.1 Calculation 1

Table 2 contains the results of the first calculation for the candidate methods described in section 4.1. The computed RBT for each method is given in appendix I.

Table 2. Results of synthetic data calculation 1 for the candidate methods.

Method	Error(R_1,R_2)	Percent Difference Norms (%)	Angle Between Vectors (milliradians)
POSIT	3.6603e-005	-0.00542	0.01779
Tsai	7.3911e-007	-0.00014	0.0000596
Bouguet	7.92894e-13	-8.1087e-11	1.49012e-05

The results shown in table 2 indicate excellent agreement between the computed RBTs and $\boldsymbol{R_S}$ and $\boldsymbol{T_S}$. In all cases, the Error($\boldsymbol{R_1}$,$\boldsymbol{R_2}$) is less than our figure of merit 0.001. The two camera calibration methods, Tsai and Bouguet, produce more accurate results than the object pose method POSIT. This is not unexpected since POSIT internally quantizes the pixels during part of its calculation. Based on these results, all three methods remain viable alternatives.

5.1.2 Calculation 2

For each candidate method, using 10 sets each of 4 through 35 points results in 320 different registration estimations. Results for Error($\boldsymbol{R_1}$,$\boldsymbol{R_2}$), percent difference in translation vector norm, and the angle between the computed RBT translation vector and $\boldsymbol{T_S}$ are presented in figures 11 through 19 for the different candidate methods. For each graph, the abscissa is the number of matching data points provided to the candidate method as input. No attempt was made to ensure that the same sets of points were entered in the different candidate methods. An examination of the figures indicates that all three methods produce results similar to those in table 2 for almost all size-matching point sets. POSIT shows the largest variability in results among the three methods. Tsai's method did not produce results for matching point sets of less than eight points because of the use of the non-coplanar calibration option. Bouguet's method produced the largest differences of any method for matching point sets of size 4 and 5 but had the lowest overall differences for all matching point sets of size 6 and above. Based on these results, it would appear that the camera calibration methods are slightly better than the POSIT method for input matching data point sets containing a small number of points. However, it is noted that the POSIT results remain acceptable for matching point set sizes above 5 (see figure 11). The

POSIT results simply exhibit greater variability than the results from the camera calibration methods. As with the first calculation, all methods would appear to remain viable candidates.

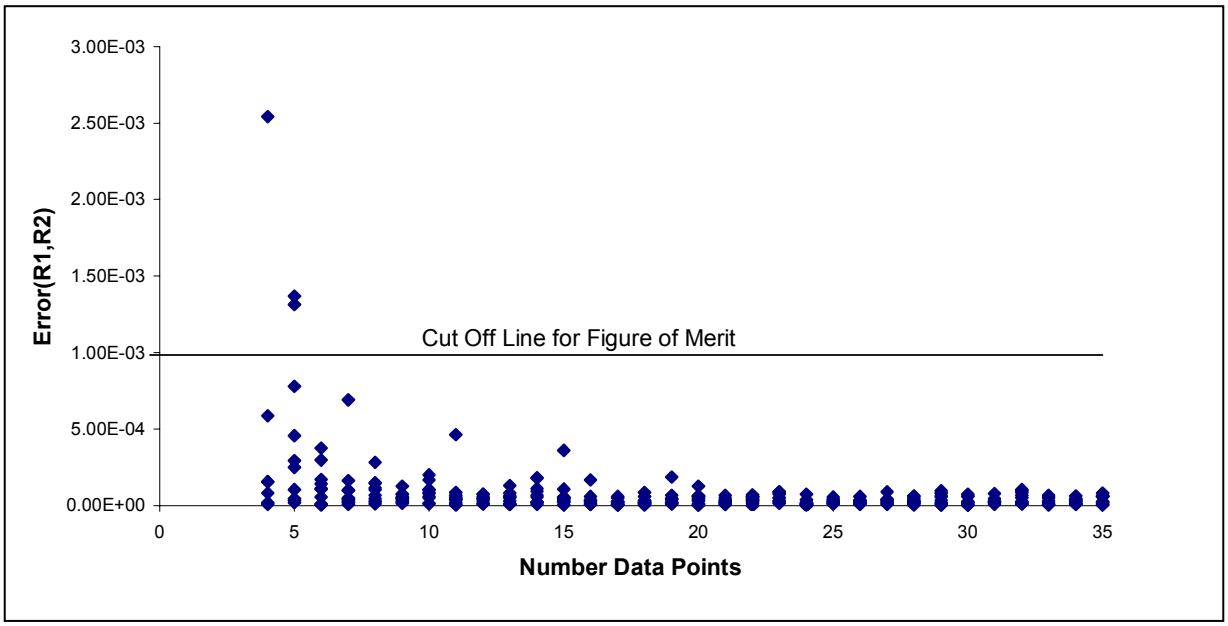

Figure 11. Error(R1,R2) for POSIT calculations, synthetic data calculation 2.

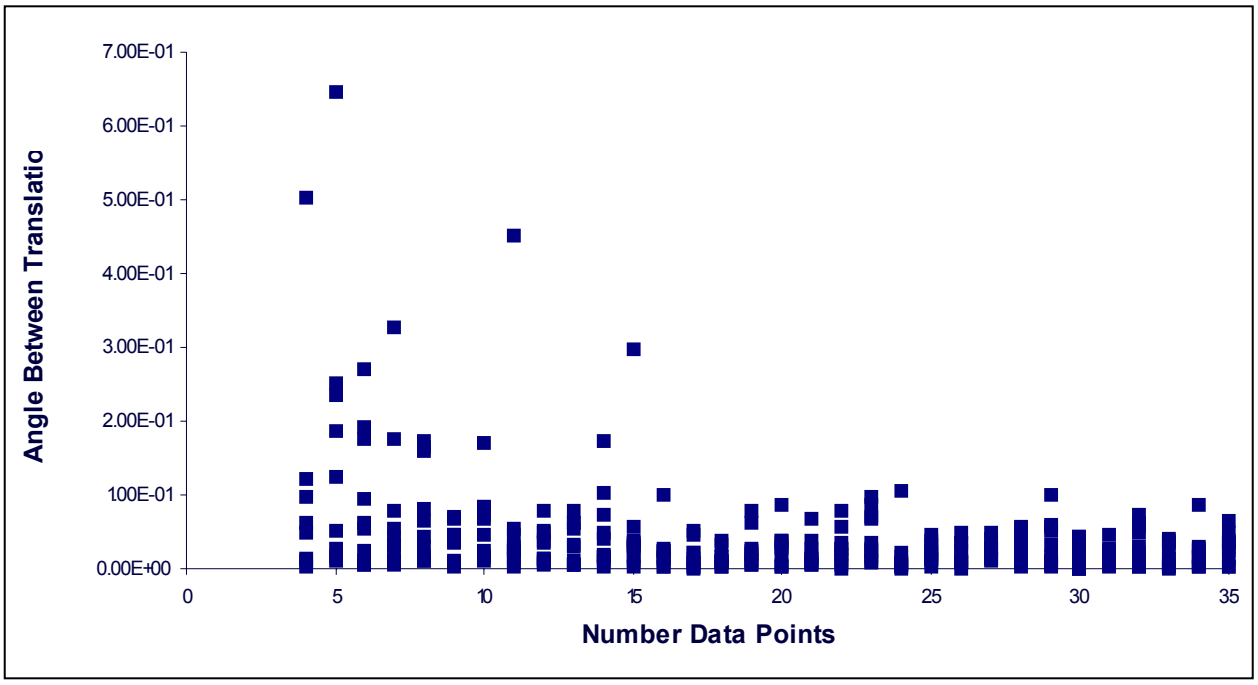

Figure 12. Angle between translation vector and Ts for POSIT calculations, synthetic data calculation 2.

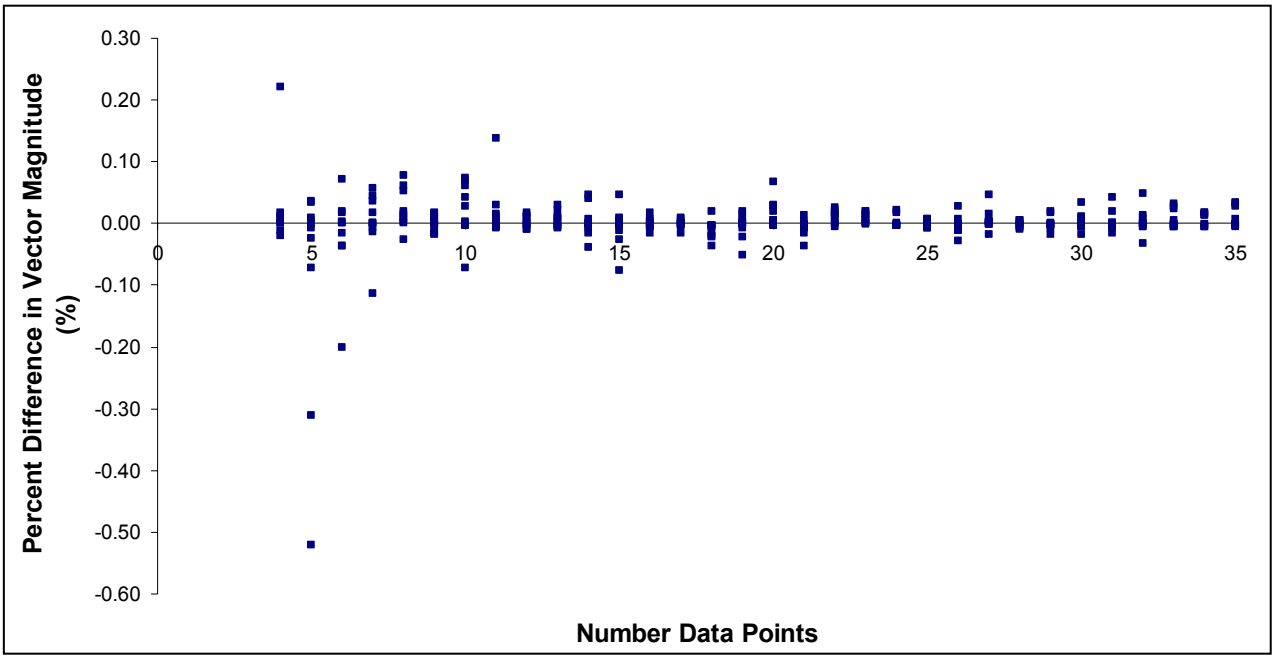

Figure 13. Percent difference in vector norm between translation vector and Ts for POSIT calculations, synthetic data calculation 2.

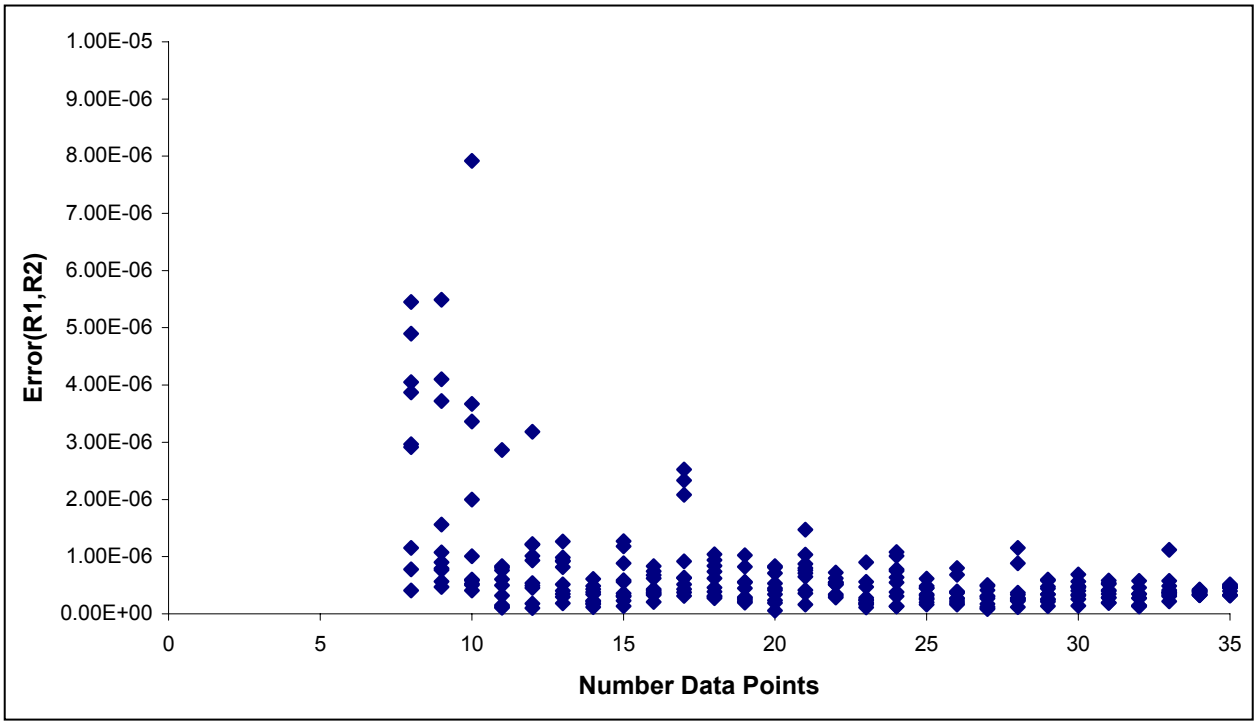

Figure 14. Error(R1,R2) for Tsai calculations, synthetic data calculation 2.

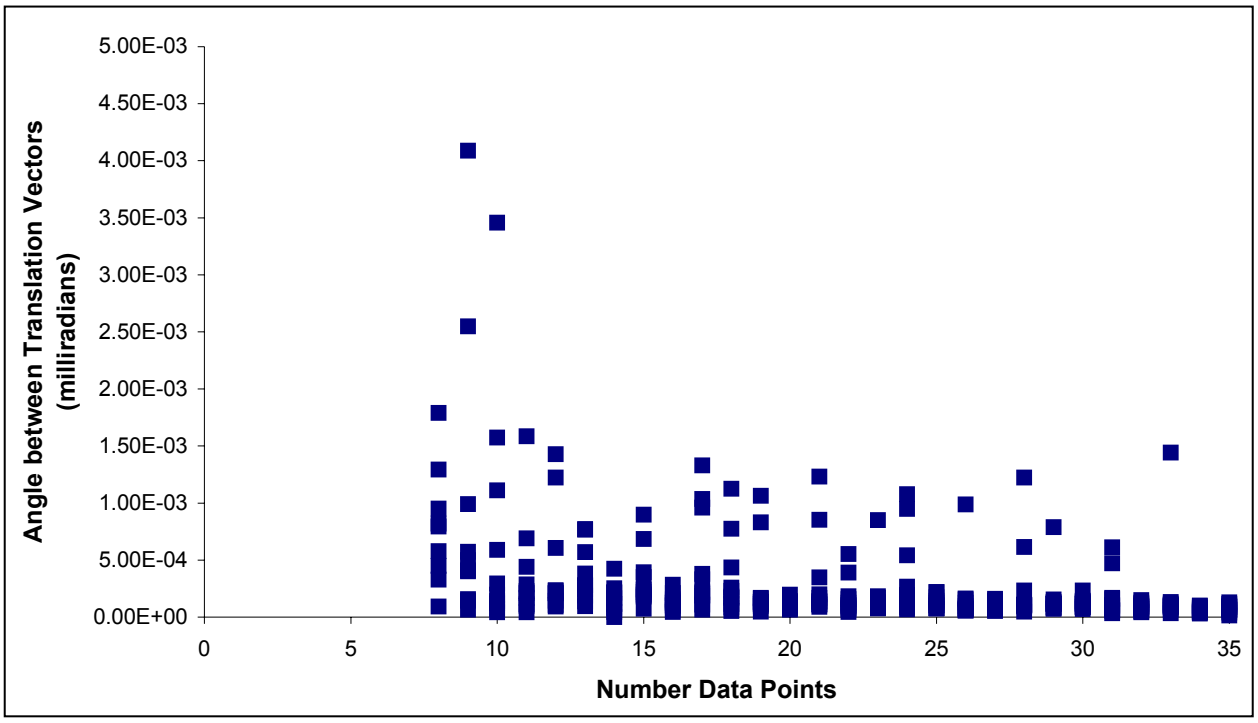

Figure 15. Angle between translation vector and Ts for Tsai calculations, synthetic data calculation 2.

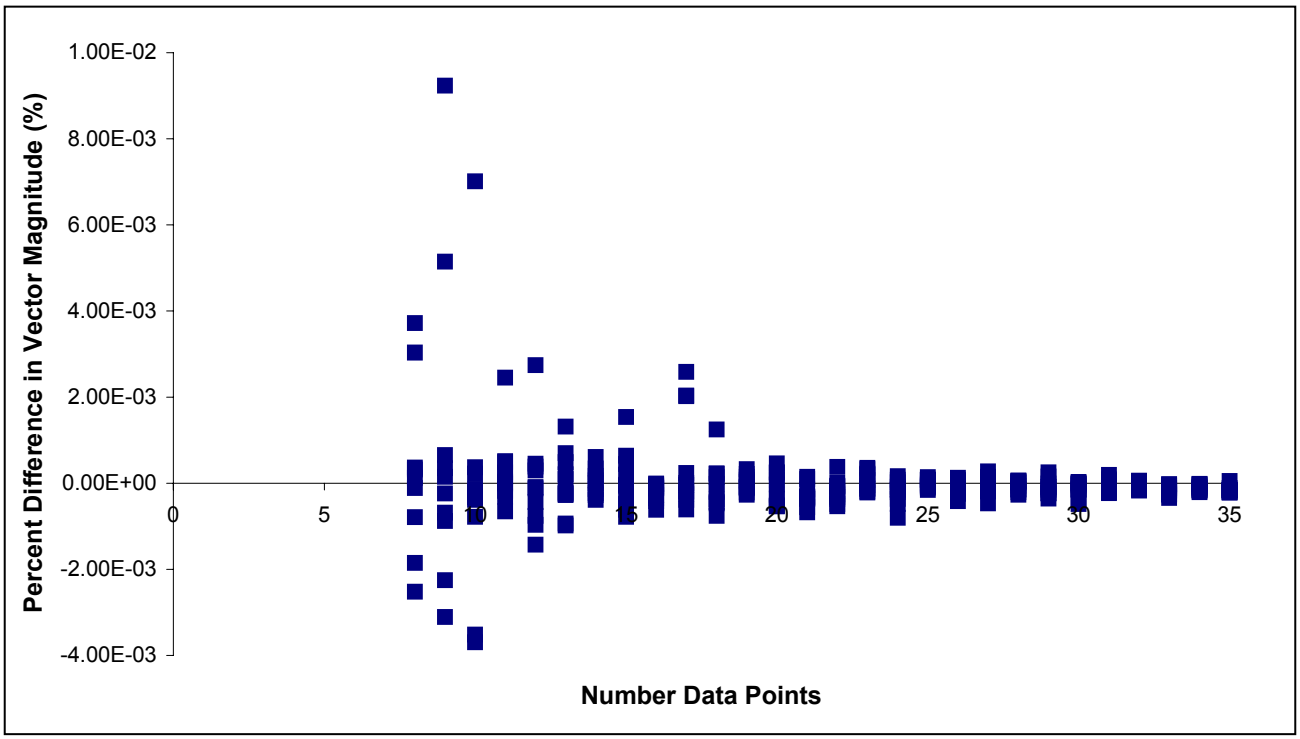

Figure 16. Percent difference in vector norm between translation vector and Ts for Tsai calculations, synthetic data calculation 2.

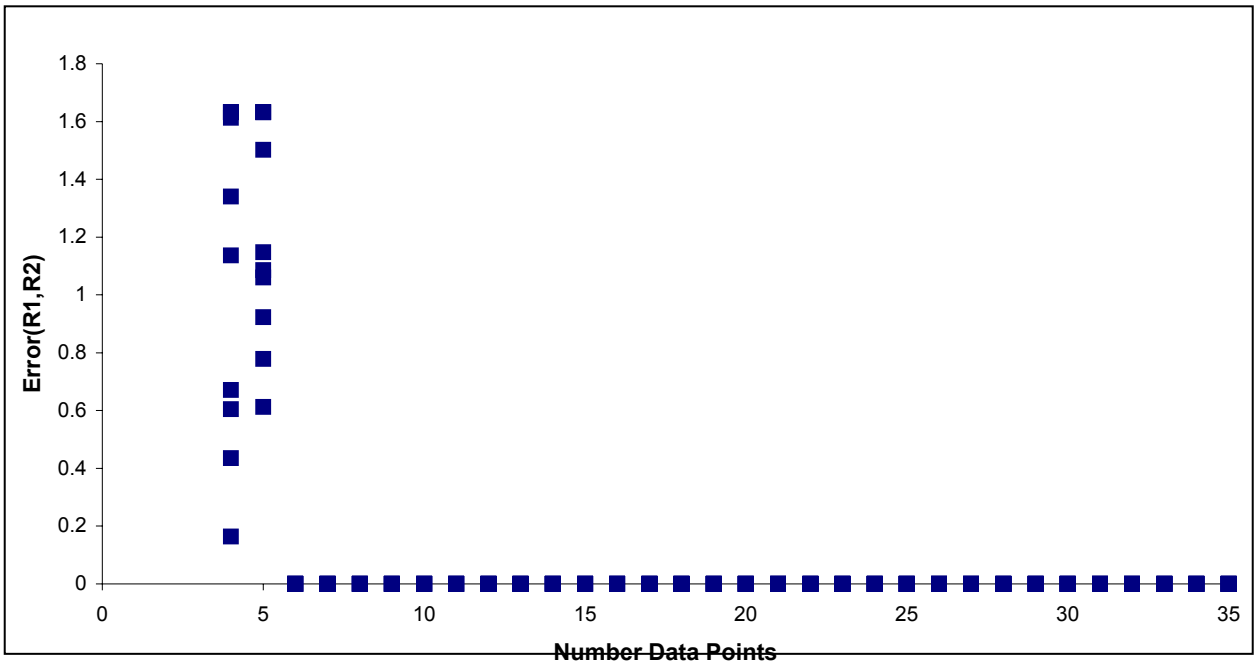

Figure 17. Error(R1,R2) for Bouguet calculations, synthetic data calculation 2.

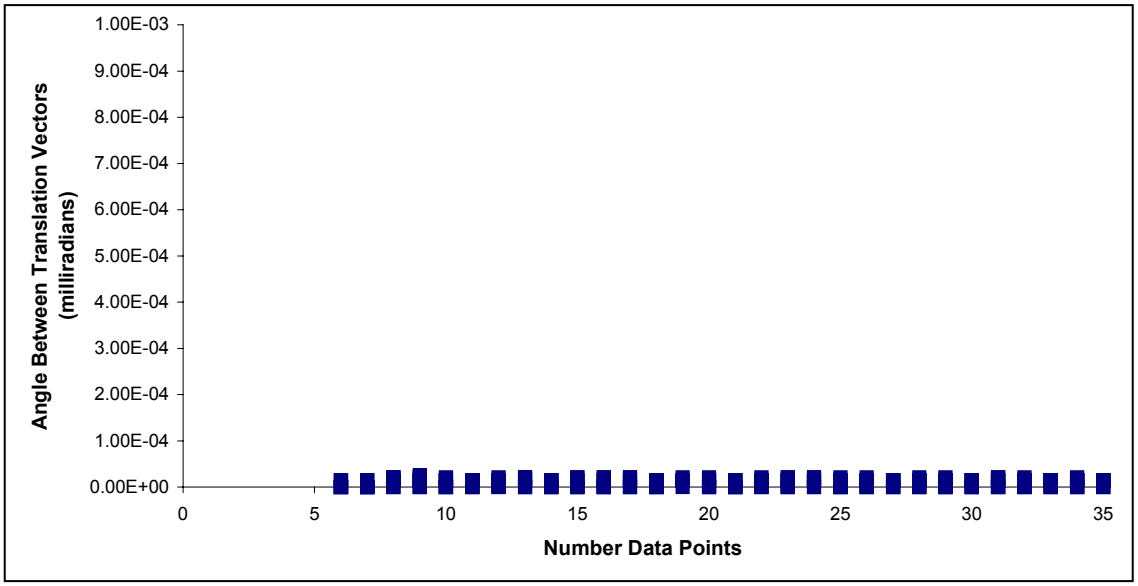

Figure 18. Angle between translation vector and Ts for Bouguet calculations, synthetic data calculation 2.

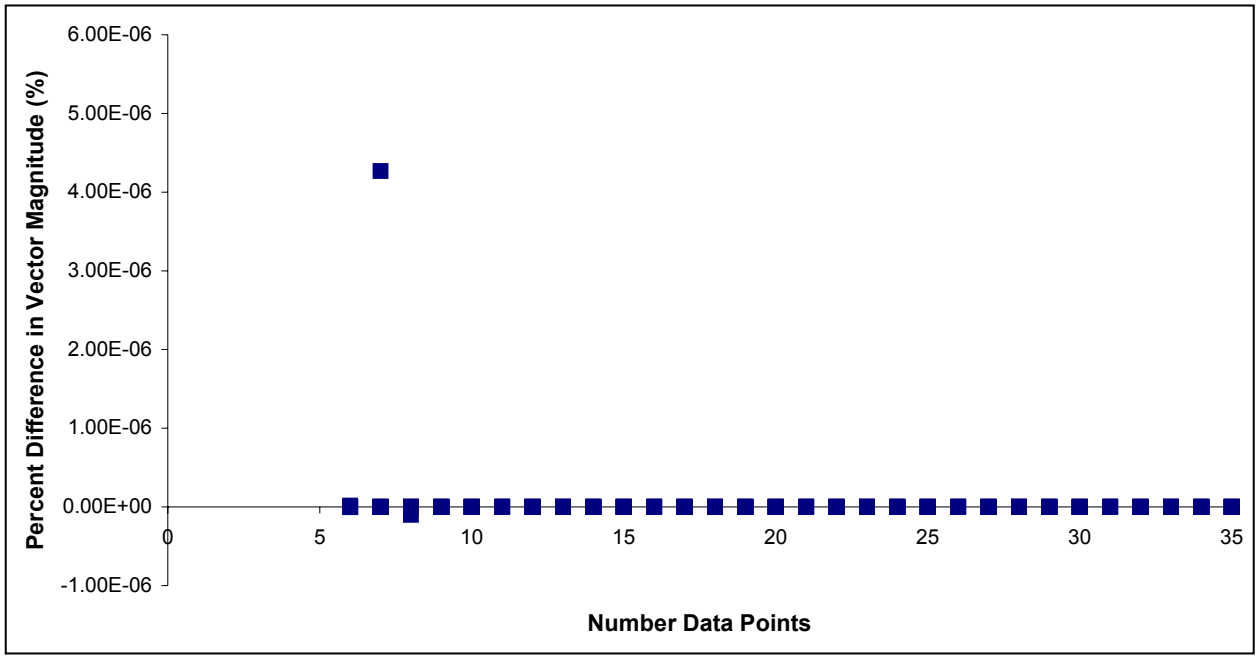

Figure 19. Percent difference in vector norm between translation vector and Ts for Bouguet calculations, synthetic data calculation 2.

5.1.3 Calculation 3

Table 3 contains the results for the third calculation for the candidate methods with the synthetic data as described in section 4.1. The computed RBT for each method is given in appendix I.

Table 3. Results for calculation 3 with synthetic data.

Method	Error(R_1,R_2)	Percent Difference Norms (%)	Angle Between Vectors (milliradians)
POSIT	0.00196351029138	0.56183452476	0.804915808
Tsai	0.00220136645337	-2.052962869152	2.120393461
Bouguet	0.00849822281720	-1.01400487131609	0.49919040174060

The only change in the input data for this calculation was the rounding of the image pixel points. This introduces an average error of approximately 0.25 pixel in the horizontal and vertical pixel values of each of the 36 2-D image points. Although the results in table 3 show that all three candidate methods produced poorer results compared to the error-free input data (table 2), the POSIT results did not degrade as much as the two camera calibration methods. This greater sensitivity of the camera calibration methods to noisy input data is consistent with the earlier comment by Faugeras (1993) and the observations of other researchers (e.g., Tapper, McKerrow, and Abrantes, 2002). The results of calculation 3 do not provide any reason to eliminate any of the candidate methods. However, the relatively large increase in errors for the camera calibration methods is worrisome.

5.2 Experimental Data Results

5.2.1 Comparison With R_{LR} and T_{LR}

Table 4 contains the results of the comparison between the calculated transformation between the left and right cameras, and the transformation defined by $\boldsymbol{R_{LR}}$ and $\boldsymbol{T_{LR}}$ as described in section 4.2. The computed RBT for each method is given in appendix J.

Table 4. Results of the comparison for the different camera registrations with R_{LR} and T_{LR}.

Method	Error(R_1,R_2)	Percent Difference Norms (%)	Angle Between Vectors (milliradians)
POSIT	0.01204432894	-8.454	839.944
Tsai	1.018288745457	-94.747	1456.860
Bouguet	0.13158480563191	-76.143	1404.218

Upon initial inspection of the results in table 4, it would appear that none of the results for the candidate methods compare well with the transformation defined by $\boldsymbol{R_{LR}}$ and $\boldsymbol{T_{LR}}$. POSIT performs best among the candidates. Given the fact that $\boldsymbol{R_{LR}}$ and $\boldsymbol{T_{LR}}$ only approximately defines the registration between the cameras, the POSIT results would be considered good except for the difference of almost 840 milliradians (48 degrees) between the translation vectors. However, DeMenthon (April 2004) notes that the magnitude of the angle between the translation vectors may be misleading in situations when the distance between the camera centers (i.e., camera baseline) is small compared to the distance to scene elements. Assuming that $\boldsymbol{T_{LR}}$ is a reasonable approximation to the true translation vector, the distance between the camera centers is roughly 0.334788 meter. Referring to appendix B, the majority of scene elements (32 of the 36 LADAR points) are between approximately 7.5 and 17.5 meters from the sensors. Consider figure 20 which schematically portrays the experimental configuration of the stereo cameras and two possible translation vectors. In this situation, the effective error in the estimated depth for the cameras is given by 0.334788 * tan(.839944) = 0.373458 meter. This represents only 2% to 5% of the depth of scene elements. Based on this result, the POSIT calculation compares favorably with $\boldsymbol{R_{LR}}$ and $\boldsymbol{T_{LR}}$. The unknown in the calculation is the error in the input data since the matching of points was manually performed. However, at most there should be no more than a 1- to 2-pixel error in the image pixel coordinates. The Riegl LMS Z210 documentation indicates 3-D location errors of less than 20 parts per million.

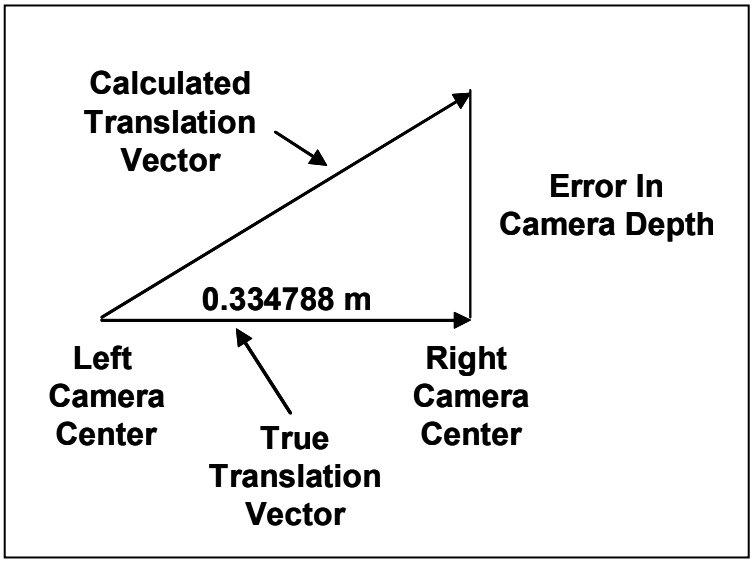

Figure 20. Schematic of experimental stereo camera configuration and two possible translation vectors.

5.2.2 Comparison With Image Pixel Locations

The results for the projection of the 3-D LADAR onto the left and right camera images are shown in table 5 for the mean and standard deviation and in figures 21 through 23. Figures 21 and 22 are for the projections with RBTs computed by POSIT. Except for the four points in the foreground and lower right corner of the pink square of the calibration stand, the POSIT projections onto the left and right images agree well with the input pixel locations. These five points contribute significantly to the standard deviations in table 5 for the horizontal and vertical pixel differences observed in the left and right camera images. On the other hand, the Tsai and Bouguet methods produce extremely poor results. The means and standard deviations in table 5 for these methods are very large. Figure 23 shows the projection of the 3-D LADAR points onto the left camera image for the Tsai RBT. As can be observed, only 29 of the 36 points even project onto the image, and these are not close to the actual pixel locations. No points were projected onto the right camera image with the Tsai RBT or onto either camera image with the Bouguet RBTs. Based on these results, the POSIT method would appear to be the best method to use in attempting to register LADAR sensors and cameras.

Table 5. Statistical results of projecting the 3-D LADAR points onto the left and right camera images with the RBTs computed by the candidate methods.

Method	Left Camera				Right Camera			
	Horizontal		Vertical		Horizontal		Vertical	
	Mean (pixels)	Standard Deviation (pixels)	Mean (pixels)	Standard Deviation (pixels)	Mean (pixels)	Standard Deviation (pixels)	Mean (pixels)	Standard Deviation (pixels)
POSIT	-5.39	12.54	0.045	20.44	3.63	13.12	-4.74	16.99
Tsai	-214.29	168.31	37.86	34.63	-1940.1	10470	539.73	2661
Bouguet	-70694	487190	-89962	610429	5635	8178	7501	11752

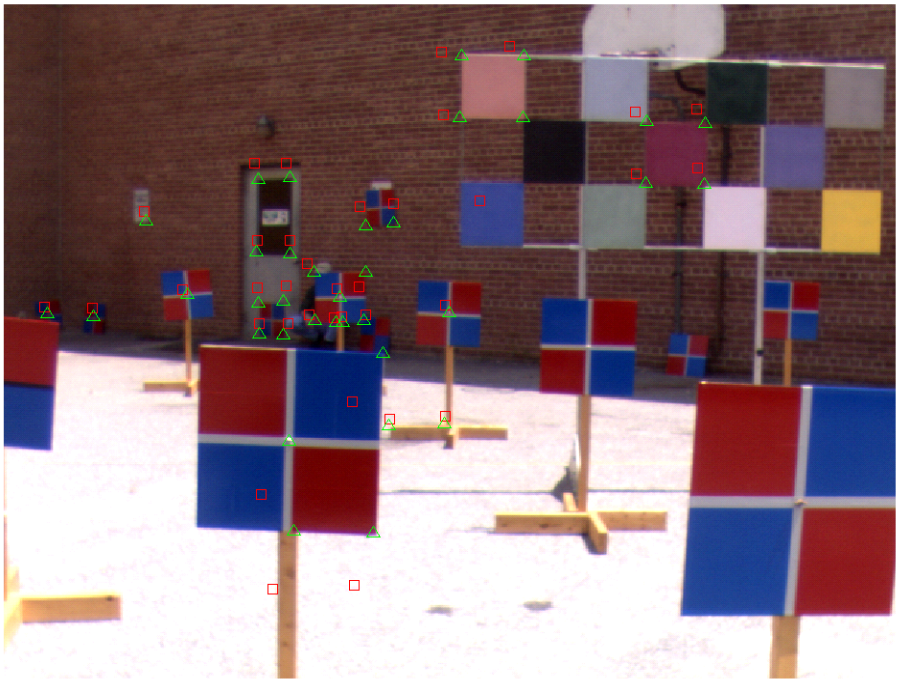

Figure 21. Projected LADAR points (red square) via POSIT-calculated transformation between LADAR and left camera; pixel locations used in calculation also shown (green triangle).

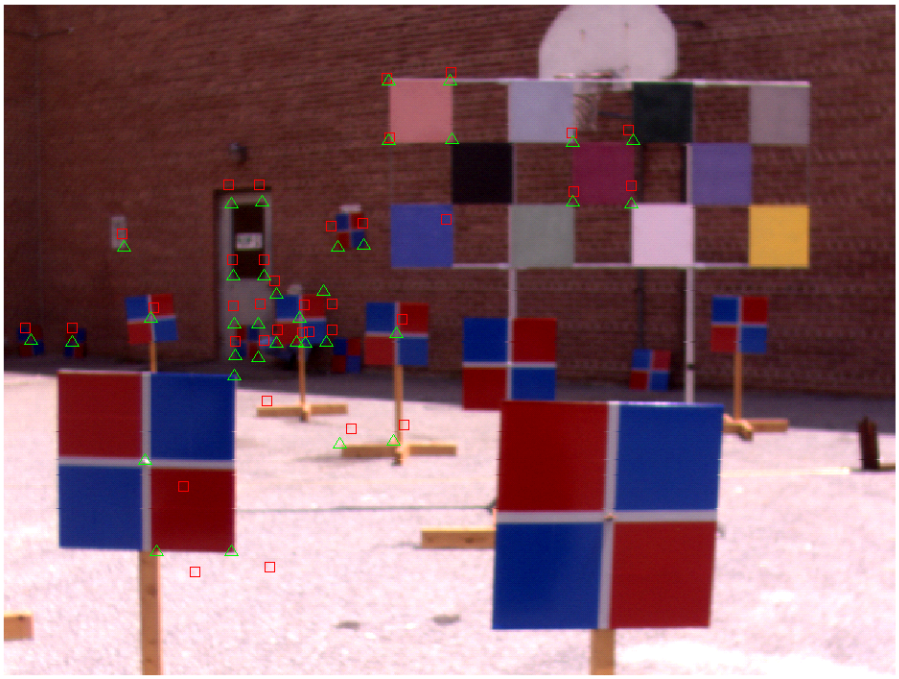

Figure 22. Projected LADAR points (red square) via POSIT-calculated transformation between LADAR and right camera; pixel locations used in calculation also shown (green triangle).

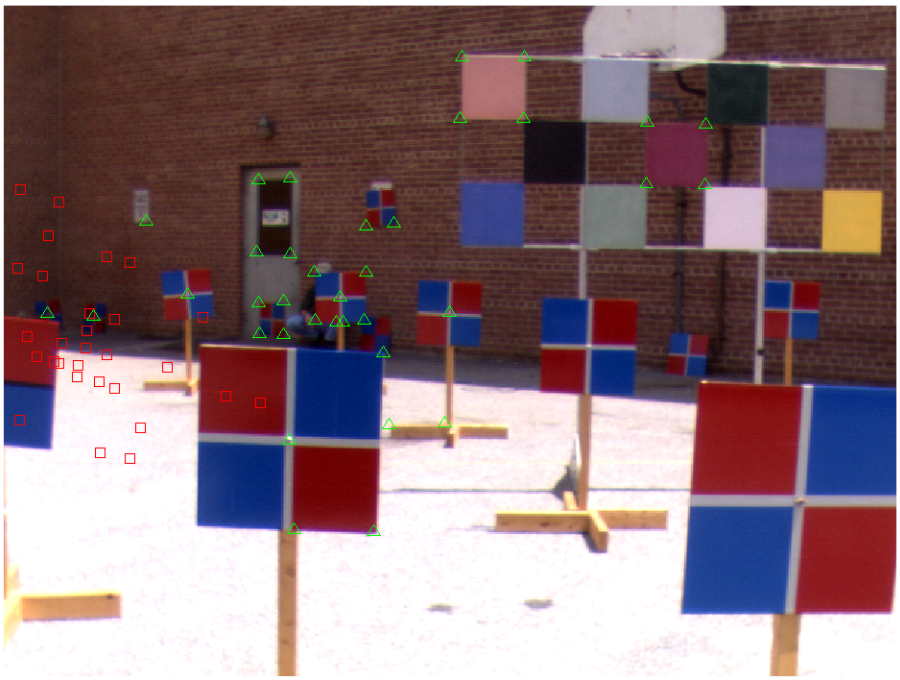

Figure 23. Projected LADAR points (red square) with Tsai-calculated transformation between LADAR and left camera; pixel locations used in calculation also shown (green triangle).

6. Discussion and Conclusions

The focus of this report is to identify potential methods to register LADAR sensor and camera information in order to facilitate the fusion of the LADAR and camera data. Since LADAR data are 3-D and camera data are 2-D, the registration problem in this case is conceptually equivalent to both the object prose and camera calibration problems. Fortunately, both the object pose and camera calibration problems have been heavily studied and discussed in the computer vision literature. Drawing on this body of work, we selected three methods representing work in these areas to determine their applicability to the LADAR-camera registration problem. The methods are an object pose method, POSIT (DeMenthon, 1995), and two camera calibration methods, one by Tsai (1986, 1987) and the other by Bouguet (2003).

Both synthetically generated (error free with ground truth) and experimental data without ground truth are used to evaluate the three methods. Results for all three methods involving the synthetically generated data compared favorably with the ground truth. However, it does appear, based on the limited calculations performed in the report, that the camera calibration methods are more sensitive to noisy input data.

Except for the POSIT calculation, results for the experimental data are far less favorable. None of the camera calibration methods appeared to predict well the translation vector for the registration. The poor results of the projections of the 3-D LADAR points onto the left and right camera images for the Tsai and Bouguet methods are especially surprising, given their performances as documented in the vision literature. To address this concern, the orientation of the LADAR and camera coordinate systems is investigated. As shown in figure 3, the negative x-axis of the LADAR coordinate system is the axis along which depth is measured. Generally, depth is measured along the z-axis, and division by the z-component of the 3-D point is used to convert to homogeneous image coordinates. To investigate if the orientation of the LADAR coordinate system axes is the source of the poor performance of the Tsai and Bouguet methods, additional calculations are performed. Specifically, the experimental data calculations are repeated with the 3-D LADAR data rotated (basis vectors re-ordered) so that the axis orientation is roughly that of the camera data. Each 3-D LADAR data point is multiplied by the rotation matrix:

$$\tilde{\mathbf{R}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix}.$$

In the Bouguet calculations, this is equivalent to using $\tilde{\mathbf{R}}$ and the vector $\tilde{\mathbf{T}} = (0 \;\; 0 \;\; 0)$ as the initial RBT estimate. The results of these calculations, together with the POSIT results with no rotation of the LADAR data, are provided in tables 6 and 7. The Tsai calculations for the LADAR to right camera produced unstable results (different results for repeated calculations with the same input, error message: *possible handedness problem with data* reported for several of the calculations). Appendix K lists the RBT from the POSIT and Bouguet calculations. Projections of the LADAR points onto the left and right camera images for the Bouguet calculations are given in figures 24 and 25. The projections for the POSIT calculations are the same as with the non-rotated LADAR data and are shown in figures 21 and 22.

Table 6. Results of the comparison for the different camera registrations with R_{LR} and T_{LR} for the calculations using the rotated LADAR data.

Method	Error(R_1,R_2)	Percent Difference Norms (%)	Angle Between Vectors (milliradians)
POSIT Non-Rotated LADAR Data	0.01204432894	-8.454	839.944
POSIT Rotated LADAR Data	0.01204432894	-8.454	839.944
Tsai Rotated LADAR Data	Unstable Results	Unstable Results	Unstable Results
Bouguet Rotated LADAR Data	0.00432979323664	1.3606	23.008

Table 7. Results of the projection of the LADAR data onto the left camera image with rotated LADAR data for the candidate methods and non-rotated LADAR data for the POSIT method.

Method	**Left Camera**				**Right Camera**			
	Horizontal		**Vertical**		**Horizontal**		**Vertical**	
	Mean (pixels)	**Standard Deviation (pixels)**	**Mean (pixels)**	**Standard Deviation (pixels)**	**Mean (pixels)**	**Standard Deviation (pixels)**	**Mean (pixels)**	**Standard Deviation (pixels)**
POSIT Non-Rotated LADAR Data	-5.39	12.54	0.045	20.44	3.63	13.12	-4.74	16.99
POSIT Rotated LADAR Data	-5.39	12.54	0.045	20.44	3.63	13.12	-4.74	16.99
Tsai Rotated LADAR Data	1.06	18.52	-1.41	24.27	Unstable Results		Unstable Results	
Bouguet Rotated LADAR Data	-0.043	11.449	0.0272	15.8213	-0.0161	10.26	0.0429	15.73

The results in tables 6 and 7 clearly show the improvement in the results produced by Bouguet's method using the rotated LADAR data. In general, for UGV applications, the approximate orientation of the LADAR and camera coordinate systems is known. Thus, it should always be possible to re-order the 3-D LADAR data to align the depth along the LADAR coordinate system z-axis. Based on the results for the data set used in the paper, such an alignment of the LADAR data results in an excellent LADAR-camera registration (table 6) for Bouguet's method. Thus, Bouguet's method would appear to be the method of choice for estimating the LADAR-camera registration. If, on the other hand, the objective is to determine a projection of the LADAR data onto the left and right camera images, the results in table 7 and figures 21, 22, 24, and 25 would indicate little to choose between POSIT and Bouguet's method. Since the POSIT calculation did not require pre-processing of the LADAR data, it would appear to be the better choice between the two methods. However, even in this situation, it is recommended that Bouguet's method (with suitably aligned LADAR data) be employed with results compared to the POSIT results.

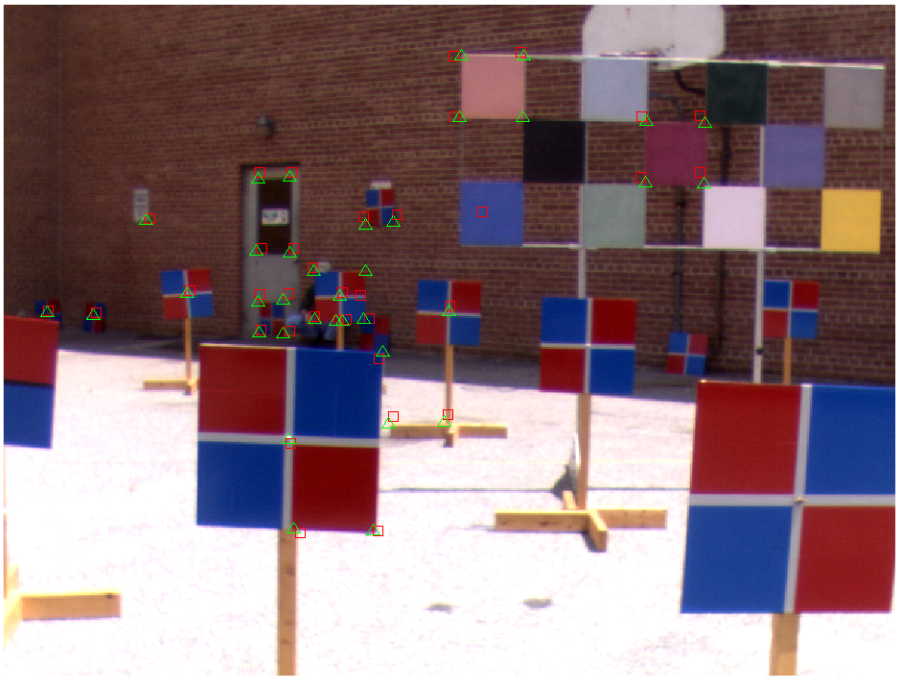

Figure 24. Projected LADAR points (red square) with Bouguet calculated transformation with rotated LADAR data between LADAR and left camera; pixel locations used in calculation also shown (green triangle).

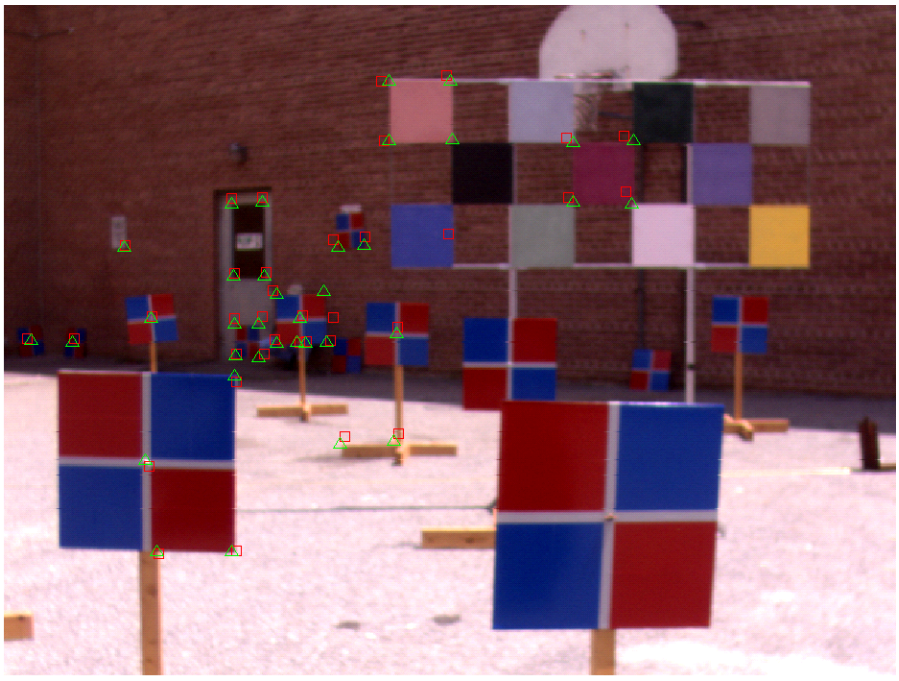

Figure 25. Projected LADAR points (red square) with Bouguet calculated transformation with rotated LADAR data between LADAR and right camera; pixel locations used in calculation also shown (green triangle).

7. References

Abidi, M. A; Chandra, T. A New Efficient and Direct Solution for Pose Estimation Using Quadrangular Targets: Algorithm and Evaluation. *IEEE Transactions on Patter Analysis and Machine Intelligence* **May 1995**, *17* (5), 534–538.

Baeten, J. *Camera Calibration Under Matlab*, Tsai camera calibration code software. Katholieke Universiteit Leuven, Belgium, 2003. Matlab code available online at web site: www.khlim.be/~jbaeten/star/camcalE.html.

Bouguet, J.-Y. *Camera Calibration Toolbox for Matlab*, web site: http://www.vision.caltech.edu/bouguetj/, 2003.

Cobzas, D.; Zhang, H.; Jagersand, M. A Comparative Analysis of Geometric and Image-Based Volumetric and Intensity Data Registration Algorithms. *Proc. of IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington DC, May 2002.

DeMenthon, D. F., Private Communications, University of Maryland, College Park, Maryland, January 2004.

DeMenthon, D. F., Private Communications, University of Maryland, College Park, Maryland, April 2004.

DeMenthon, D. F.; Davis, L. S. Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision* **1995**, *15*, 123–141.

Faugeras, O. *Three-Dimensional Computer Vision: A Geometric Viewpoint*; The MIT Press: Cambridge, Massachusetts, 1993.

Fischler, M.; Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the Association for Computing Machinery (ACM)* **June 1981**, *24* (6), 381–395.

Hanek, R.; Navab, N.; Appel, M. Yet Another Method for Pose Estimation: A Probabilistic Approach using Points, Lines and Cylinders. *IEEE Conference on Computer Vision and Pattern Recognition* **1999**, *II*, 544–550.

Haralick, R. M.; Lee, C.; Ottenberg, K.; Nolle, M. Analysis and Solutions of the Three Point Perspective Pose Estimation Problem. *Proceedings IEEE Conference of Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991; 592–598.

Horaud, R.; Conio, B.; Leboulleux, O. An Analytic Solution for the Perspective 4-Point Problem. *IEEE Computer Vision, Graphics, and Image Processing* **1989**, *47*, 33–44.

Manning, R. A. Screw-Transform Manifolds for Camera Self Calibration. Ph.D. Thesis, University of Wisconsin-Madison, 2003.

Oberle, W. F. *The Effect of Variability in Stereo Camera Calibration and Registration Parameters on Three-Dimensional Reconstruction Distance Estimates*; ARL-TR-3140; U. S. Army Research Laboratory: Aberdeen Proving Ground, Maryland, February 2004.

Quan, L.; Lan, Z. Linear N-Point Camera Pose Determination. *IEEE Transactions on Pattern Analysis and machine Intelligence* **July 1999**, *21* (7), 774–780.

Riegl Laser Measurement Systems GmbH, Horn, Austria, web site: www.riegl.com.

Tapper, M.; McKerrow, P. J.; Abrantes, J. Problems Encountered in the Implementation of Tsai's Algorithm for Camera Calibration. *Proceedings 2002 Australasian Conference on Robotics and Automation*, Auckland, 27–November 2002, 66 – 70.

The MathWorks, Inc. MATLAB, Version 6.1.0.450, Release 12.1, web site: http://www.mathworks.com, 2001.

Trucco, E.; Verri, A. *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, Inc., Upper Saddle River, New Jersey, 1998.

Tsai, R. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, 364 – 374,.

Tsai, R. A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation* **August 1987**, *RA-3* (4), 323–344.

Willson, R. Tsai camera calibration code software. Carnegie Mellon University, Pennsylvania, 1995. C code available online at web site: www-2.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html.

Wald L. Some Terms of Reference in Data Fusion. *IEEE Transactions on Geoscience and Remote Sensing* **May 1999**, 37 (3), 1190.

Appendix A. C++ Code for Mapping Original Intensity Image Pixels to Rotated Intensity Image Pixels

```
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdlib.h>
#include <iomanip>
using namespace std;

void main()
{
      double x,y,z,r,a1,a2;
      long int i,red,b,g,index;
      char cLine[160];
      int iXO,iYO,iXR,iYR;
      ifstream input(<Original LADAR data file>,ios::in | ios::binary);
      ofstream output(<Output file>, ios::out | ios::binary);

      for (int jj=0; jj<8;jj++) // Read and discard header information
      {
            input.getline(cLine,159,'\n');
      }

      output << "All information is relative to the upright image. X axis
      away  from image:\n";
      output << "Y axis to right: Z axis upward\n";
      output << setw(10) << "x" << setw(10) << "y" << setw(10) << "z" <<
      setw(10) << "range" << setw(10) << "x(image)" << setw(10) << "y(image)
      \n";

        WHILE (INPUT >> X >> Y >> Z >> R >> I >> RED >> B >> G >> A1 >> A2 >> INDEX)
      {
            cout << index;
            iXO = index % 536; // Compute x coordinate in original image
            iYO = floor((index - iXS)/536); // Compute y coordinate in
            original image

            iXR=615-iYO; // Compute x coordinate in rotated/upright image
            iYR=iXO; // Same for y coordinate in rotate/upright image

            output << setw(10) << x << setw(10) << y << setw(10) << z <<
            setw(10) << r << setw(10) << iXR << setw(10) << iYR << endl;
      }

      input.close();
      output.close();
}
```

INTENTIONALLY LEFT BLANK.

Appendix B. Matched Points Riegl LADAR and Stereo Camera Pair

LADAR Pixel		LADAR 3-D Location (m)			Left Camera Image		Right Camera Image	
X	Y	x	y	z	X	Y	X	Y
156	120	-17.557	-4.845	1.009	103	155	87	174
238	141	-16.413	-2.754	0.480	206	178	187	195
234	167	-16.525	-2.851	-0.061	201	212	183	229
235	189	-16.527	-2.838	-0.526	201	236	183	253
218	188	-16.723	-3.235	-0.510	184	235	167	252
218	167	-16.731	-3.237	-0.061	183	213	166	229
297	125	-15.398	-1.408	0.759	280	156	259	173
278	125	-15.661	-1.813	0.775	260	158	240	174
219	96	-16.619	-3.191	1.449	183	125	164	143
237	97	-16.327	-2.758	1.397	206	124	186	142
219	140	-16.696	-3.206	0.514	182	177	165	195
313	95	-7.681	-0.541	0.667	329	37	277	55
347	95	-7.586	-0.209	0.659	374	37	321	55
347	127	-14.699	-0.428	0.686	373	81	322	97
313	126	-7.709	-0.543	0.368	328	81	277	98
439	129	-7.643	0.680	0.337	462	84	409	99
439	158	-7.651	0.680	0.058	504	85	452	98
409	159	-7.711	0.391	0.048	503	129	451	143
409	129	-7.640	0.388	0.336	461	128	409	142
99	170	-18.547	-6.582	-0.146	32	221	20	241
125	172	-18.271	-5.822	-0.189	65	223	50	242
177	181	-11.851	-2.924	-0.259	132	207	106	225
261	185	-11.632	-1.595	-0.308	242	209	213	225
245	170	-11.728	-1.851	-0.087	223	191	196	208
276	170	-16.239	-1.920	-0.121	260	191	230	206
276	199	-11.812	-1.390	-0.521	259	226	232	242
245	199	-11.496	-1.814	-0.509	224	226	196	243
263	200	-11.715	-1.578	-0.532	244	227	217	243
259	200	-11.755	-1.644	-0.536	239	227	211	242
316	208	-9.429	-0.634	-0.522	320	220	282	236
314	266	-9.523	-0.664	-1.225	317	299	280	313
284	266	-9.555	-1.024	-1.233	277	301	242	315
214	343	-4.380	-0.853	-1.015	205	312	102	327
248	379	-4.378	-0.658	-1.218	266	377	164	392
217	378	-4.350	-0.830	-1.214	209	376	110	392
250	310	-4.421	-0.652	-0.825	273	249	166	266

INTENTIONALLY LEFT BLANK.

Appendix C. C++ Code to Read Data for the Rotated Intensity Image (created with the program in Appendix A) and to Determine 3-D Locations

```
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdlib.h>
#include <iomanip>

using namespace std;

void main()
{
	char cBuffer[100];
	int iCount = 0, iFlag, i, j, iCount1 = 0;
	double data[100][9] = {0}, dJunk[6];
	bool bFlag = true;

	ofstream OutPutData(<Output File Name>,ios::out|ios::binary);

	while(bFlag)
	{
		cout << "Enter Riegl Image X Y:\n|------->>";
		cin >> data[iCount][0] >> data[iCount][1];
		cout << "Enter Left Image X Y:\n|------->>";
		cin >> data[iCount][5] >> data[iCount][6];
		cout << "Enter Right Image X Y:\n|------->>";
		cin >> data[iCount][7] >> data[iCount][8];
		iCount++;
		cout << "Continue? |------->>\n";
		cin >> iFlag;
		if(iFlag == 0) bFlag = false;
	}

	ifstream ReadData(<Input Data File>,ios::in | ios::binary);
	for(i = 0; i < 3; i++)
		ReadData.getline(cBuffer,sizeof(cBuffer));
	for(i = 0; i < 284682; i++)// There are 284682 lines of data
	{
		cout << "WORKING READING LINE: " << i << endl;
	ReadData >> dJunk[0] >> dJunk[1] >> dJunk[2] >> dJunk[3] >> dJunk[4] >>
	dJunk[5];
		for(j = 0; j < iCount; j++)
		{
			bFlag = false;
			if( data[j][0] == dJunk[4] && data[j][1] == dJunk[5])
			{
				data[j][2] = dJunk[0];
				data[j][3] = dJunk[1];
				data[j][4] = dJunk[2];
				iCount1++;
				bFlag = true;
			}
```

```
                if (bFlag) break;
            }
            if(iCount == iCount1) break;
        }

        ReadData.close();

        for(i = 0; i < iCount; i++)
        {
            for(int j = 0 ; j < 9; j++)
            {
                OutPutData << data[i][j] << "\t";
            }
            OutPutData << endl;
        }
        OutPutData.close();
}
```

Appendix D. MATLAB Function to Map 3-D Points to Pixel Coordinates RBA Format

```
function [CamPixel]= MapPts(rotNew1,transNew1,LadarPts,focalLength,centerLocation)

% This function will compute the pixel locations of LadarPts under the
% rigid-body transformation defined by (rotNew1,transNew1) for a camera
% with the focalLength and camera centerLocation. The assumption is that
% the rigid-body transformation is defined by pt(camera) =
% rotNew1*pt(Ladar)+transNew1

r0 = rotNew1 * LadarPts';

NumPts = length(LadarPts);

for i=1:NumPts % Translataion is added
   r1(1,i) = r0(1,i) + transNew1(1);
   r1(2,i) = r0(2,i) + transNew1(2);
   r1(3,i) = r0(3,i) + transNew1(3);
end

for i=1:3
   for j=1:NumPts
      r2(i,j) = r1(i,j)/r1(3,j); % Divide by Z coordinate
   end
end

r3 = r2 .*focalLength;

r4 = r3(1:2,:); % Select pixel coordinates

for j = 1:NumPts
   r5(1,j) = r4(1,j)+centerLocation(1);
   r5(2,j) = r4(2,j)+centerLocation(2);
end

CamPixel = r5; % Return pixel coordinates
```

INTENTIONALLY LEFT BLANK.

Appendix E. Synthetic Pixel Coordinates

1.0e+002 *

1.57740805375936	2.62013720111079
2.11894457809250	2.77628643947216
2.08801300881948	2.92469652490331
2.08671272940100	3.05298895046885
1.98035686146655	3.04427458696963
1.98477891560029	2.92111014475443
2.49126644629362	2.70613220452927
2.37772860358684	2.69929769711073
2.01154553183972	2.50748582644764
2.12625149304475	2.52260415161046
1.99883626877053	2.76385718297269
2.78317368772487	2.69385351425107
2.91001273192363	2.70033132558568
2.77316862153503	2.73270680529859
2.77827121881786	2.80708619332384
3.24084016564146	2.83349769827049
3.23723551750734	2.93891997313893
3.12680610410456	2.93916186592248
3.13086695611266	2.83033471412728
1.13546354326037	2.91379893079620
1.32611132377672	2.93164498856752
1.97284811919576	2.99600602200180
2.40252357198268	3.02600538329852
2.32260406641536	2.95138250814487
2.34282698539017	2.94978251045506
2.46772628326252	3.09576078765975
2.32721010394602	3.09033720635560
2.40599354018993	3.09832110535040
2.38478828376005	3.09866277366173
2.72276828227229	3.12493200210836
2.70306414433192	3.37148976237305
2.57590623396992	3.37029633409025
2.64941218052097	3.40516424123719
2.73309795950656	3.49801370764481
2.65661900992197	3.49516632130007
2.74155170873795	3.32196448051251

INTENTIONALLY LEFT BLANK.

Appendix F. MATLAB Function to Compare Two RBTs

```
function[frob,angle,diffmag]=CompRBT(Rot1,Trans1,RotBase,TransBase)
% This program will compute the statistics associated with the two
% rigid-body transformations defined by (Rot1,Trans1) and
% (RotBase,TransBase). The translations are assumed to be 3-by-1 column
% vectors. See text for details. Calculations call the function
% frobenious.m which takes as an argument the matrix for which the
% frobenious norm will be determine. The return values are non-dimensional
% for frob, milliradians for angle, and % for diffmag.

% Start with the rotation matrix error.

t1 = Rot1/frobenious(Rot1);
t2 = RotBase/frobenious(RotBase);
t3 = t1-t2;
frob = frobenious(t3);

% Next the percent difference in the translation vector norms.

t1 = norm(Trans1);
t2 = norm(TransBase);
diffmag = (t1-t2)/t2*100;

% Finally the angle between the vectors is computed.

t3 = dot(Trans1,TransBase)/t1/t2; % Cosine of angle between vectors
angle = acos(t3)*1000;



function[norm]=frobenious(a)
norm = sqrt(sum(sum(a.*a)));
```

INTENTIONALLY LEFT BLANK.

Appendix G. MATLAB Script File to Select Random Subsets and Perform a Comparison

```
% This MATLAB script is ComparisonPosit.m and is used to perform
% calculation 2 for the synthetic data as described in the body of the
% text. Will select 10 each subsets of size 4 through 35.

focalLength = focalLeft; % Left camera intrinsic data used in generating
synthetic data
centerLocation = centerLeft;
fid=fopen('ComResultsPositRT.txt','w'); % Need to save certain information to
files
fid1=fopen('ComResultsPosit.txt','w'); % Error statistic only
NumPts = 36; % 36 data points in synthetic data

for i=4:35% Range of point subset size
    for i1 = 1:10 % Going to do each subset 10 times.
        index =randperm(NumPts);  % First need to get the indices for the
            desired points
        % Get the data from the image and Ladar data
        array1 = PixelSD(index(1:i),:);
        array2 = LadarData(index(1:i),:);
        % Run the appropriate method for the data set (in this case
        % classicPosit

[rotNew,transNew,cbillNew]=classicPosit(array1,array2,focalLength,centerLocat
ion);
    if (sum(sum(isnan(rotNew))) > 0) % Check to see if invalid calculation
        continue
    end
    [u,s,v]=svd(rotNew); % Use SVD to ensure result is orthogonal
    rotNew = u*v';
    % Compute error statistics and save results
    fprintf(fid,'**************\n'); % Writing out the transformation
    fprintf(fid,'Number of Points = %d\n',i);
    fprintf(fid,'%d\t%d\t%d\n',rotNew(1,1),rotNew(1,2),rotNew(1,3));
    fprintf(fid,'%d\t%d\t%d\n',rotNew(2,1),rotNew(2,2),rotNew(2,3));
    fprintf(fid,'%d\t%d\t%d\n',rotNew(3,1),rotNew(3,2),rotNew(3,3));
    fprintf(fid,'%d\t%d\t%d\n',transNew(1),transNew(2),transNew(3));

    [f,a,diff]=CompRBT(rotNew,transNew,RotSD,TransSD');
    fprintf(fid1,'%d\t%d\t%d\t%d\n',i,f,a,diff); % File #pts error(R1,R2)
angle
       diffmag
    end % This is the end of the i1 loop.
end % This is the end of i loop.
fclose(fid);
fclose(fid1);
```

INTENTIONALLY LEFT BLANK.

Appendix H. MATLAB Script File to Compare Mapping of LADAR Data to Left and Right Images Via Computed RBTs

```
% This script will use the rigid-body transformations to map a set
% of 3-D points identified below (in the present case the LADAR data) to
% the left and right images of a pair of cameras in order to compare the
% accuracy of the mapping to the known image points. The computed values
% are the mean and standard deviations of the mapped points to the known
% image points. File: ComparePts.m

% Mapping to Left image is performed
[CamPixelLeft] = MapPts(RotPL,TransPL,LadarData,focalLeft,centerLeft);
CamPixelLeft = CamPixelLeft';
% Plot the results on Image
figure
g1=imread('calWTargets2Left.ppm');
imshow(g1);
hold on
plot(CamPixelLeft(1:36,1),CamPixelLeft(1:36,2),'rs');
plot(LeftImage(1:36,1),LeftImage(1:36,2),'g^');
hold off
A=CamPixelLeft - LeftImage;
meanLeft=mean(A);
stdevLeft=std(A);

% Mapping to Right image is performed
[CamPixelRight] = MapPts(RotPR,TransPR,LadarData,focalRight,centerRight);
CamPixelRight=CamPixelRight';
% Plot the results on Image
figure
g1=imread('calWTargets2Right.ppm');
imshow(g1);
hold on
plot(CamPixelRight(1:36,1),CamPixelRight(1:36,2),'rs');
plot(RightImage(1:36,1),RightImage(1:36,2),'g^');
hold off
A=CamPixelRight - RightImage;
meanRight=mean(A);
stdevRight=std(A);
```

INTENTIONALLY LEFT BLANK.

Appendix I. Rigid Body Transformations

Calculation 1 Synthetic Data

POSIT Results:

$$\boldsymbol{R}_{\text{POSIT}} = \begin{pmatrix} 0.08748666618773 & 0.99566845370243 & 0.03147083635947 \\ -0.04585758575663 & 0.03551371098605 & -0.99831651201439 \\ -0.99510990400993 & 0.08589620685912 & 0.04876592767895 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{POSIT}} = \begin{pmatrix} -0.33455633740165 & 0.89803647081185 & 15.22704598173041 \end{pmatrix}$$

Tsai Results:

$$\boldsymbol{R}_{\text{Tsai}} = \begin{pmatrix} 0.087497 & 0.995669 & 0.031421 \\ -0.045876 & 0.035536 & -0.998315 \\ -0.995108 & 0.085908 & 0.048787 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{Tsai}} = \begin{pmatrix} -0.334484 & 0.897828 & 15.227867 \end{pmatrix}$$

Bouguet Results:

$$\boldsymbol{R}_{\text{Bouguet}} = \begin{pmatrix} 0.08749666119233 & 0.99566917244720 & 0.03142026923023 \\ -0.04587594117294 & 0.03553521160298 & -0.99831490360398 \\ -0.99510789982880 & 0.08590778646105 & 0.04878647275293 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{Bouguet}} = \begin{pmatrix} -0.33448540933926 & 0.89782925664899 & 15.22788837257698 \end{pmatrix}$$

Calculation 3 Synthetic Data

POSIT Result:

$$\boldsymbol{R}_{\text{POSIT}} = \begin{pmatrix} 0.08838084128758 & 0.99550902391960 & 0.03392064545307 \\ -0.04476928444609 & 0.03442613693955 & -0.99840400252884 \\ -0.99508795082035 & 0.08672118266350 & 0.04761083760550 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{POSIT}} = \begin{pmatrix} -0.32896356714428 & 0.91275161928253 & 15.31301908881755 \end{pmatrix}$$

Tsai Result:

$$\boldsymbol{R}_{\text{Tsai}} = \begin{pmatrix} 0.090031 & 0.995464 & 0.030761 \\ -0.046528 & 0.035057 & -0.998302 \\ -0.994852 & 0.088447 & 0.049473 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{Tsai}} = \begin{pmatrix} -0.347514 & 0.903987 & 14.913345 \end{pmatrix}$$

Bouguet Result:

$$\boldsymbol{R}_{\text{Bouguet}} = \begin{pmatrix} 0.08236590309799 & 0.99613663662717 & 0.03045749786114 \\ -0.05492915423154 & 0.03505247937352 & -0.99787479760999 \\ -0.99508725548178 & 0.08051785428635 & 0.05760407206840 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{Bouguet}} = \begin{pmatrix} -0.32497859971989 & 0.89313339023542 & 15.07334937410542 \end{pmatrix}$$

Appendix J. Calculated Transformation Between Left and Right Cameras

POSIT Result:

$$\boldsymbol{R}_{\text{POSIT}} = \begin{pmatrix} 0.99960296963107 & 0.01350158696889 & -0.02473075522642 \\ -0.01366202010454 & 0.99988663530897 & -0.00632975016614 \\ 0.02464248995963 & 0.00666510913820 & 0.99967410890188 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{POSIT}} = \begin{pmatrix} 0.20209774970646 & 0.03470469200352 & -0.22778169861526 \end{pmatrix}$$
(units in meters)

Tsai Result:

$$\boldsymbol{R}_{\text{Tsai}} = \begin{pmatrix} 0.34038590296134 & -0.36263792677361 & 0.86754318113310 \\ -0.16174787504922 & 0.88630234522256 & 0.43394213643072 \\ -0.92626943271505 & -0.28803105200366 & 0.24302891000745 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{Tsai}} = \begin{pmatrix} 1.80640315877847 & 2.12913142069132 & -17.36208036043379 \end{pmatrix}$$
(units in meters)

Bouguet Result:

$$\boldsymbol{R}_{\text{Bouguet}} = \begin{pmatrix} 0.99080254911237 & -0.08294852591746 & -0.10691047993790 \\ 0.09092714048446 & 0.99324808385759 & 0.07204511806185 \\ 0.10021259299983 & -0.08110355085536 & 0.99165500565615 \end{pmatrix}$$

$$\boldsymbol{T}_{\text{Bouguet}} = \begin{pmatrix} 0.09036524651875 & -0.19845839189344 & -0.54790590935389 \end{pmatrix}$$
(units in meters)

INTENTIONALLY LEFT BLANK.

Appendix K. RBT With Rotated LADAR Data

POSIT Result:

RBT Left to Right Camera

$$\boldsymbol{R} = \begin{pmatrix} 0.99960296963107 & 0.01350158696889 & -0.02473075522642 \\ -0.01366202010454 & 0.99988663530897 & -0.00632975016614 \\ 0.02464248995963 & 0.00666510913820 & 0.99967410890188 \end{pmatrix}$$

$$\boldsymbol{T} = \begin{pmatrix} 0.20209774970646 & 0.03470469200352 & -0.22778169861526 \end{pmatrix}$$

(units in meters)

RBT LADAR to Left & Right Cameras

$$\mathbf{R}_{\mathrm{POSIT}\rightarrow\mathrm{Left}} = \begin{pmatrix} -0.01804441204342 & 0.99979479894226 & -0.00920647609134 \\ 0.01955163523604 & -0.06370295006891 & -0.99777736380022 \\ -0.99815909851653 & -0.01818430754236 & -0.01839814094038 \end{pmatrix}$$

$$\mathbf{T}_{\mathrm{POSIT}\rightarrow\mathrm{Left}} = \begin{pmatrix} 0.28412019867384 & -0.65591294859011 & -1.80752603592876 \end{pmatrix}$$

(units in meters)

$$\mathbf{R}_{\mathrm{POSIT}\rightarrow\mathrm{Right}} = \begin{pmatrix} 0.00651548396152 & 0.99989344943916 & -0.01306285716863 \\ 0.02535246882550 & -0.08636957903306 & -0.99594053444074 \\ -0.99696264989285 & 0.00615785889964 & -0.02591250711268 \end{pmatrix}$$

$$\mathbf{T}_{\mathrm{POSIT}\rightarrow\mathrm{Right}} \begin{pmatrix} 0.11173371995390 & -0.68166055430343 & -1.58181131726884 \end{pmatrix}$$

(units in meters)

Tasi Result: Unstable results for LADAR to Right Camera – Error message: possible handedness problem in data

Bouguet Result:

RBT Left to Right Camera

$$\mathbf{R} = \begin{pmatrix} 0.99971179944811 & 0.01873729760959 & -0.01500772209572 \\ -0.01881724903192 & 0.99980939677545 & -0.00520396563945 \\ 0.01490735332256 & 0.00548486989776 & 0.99987383555083 \end{pmatrix}$$

$$\mathbf{T} = \begin{pmatrix} 0.33915017084333 & 0.00062363907929 & -0.01143244829854 \end{pmatrix}$$

(units in meters)

RBT LADAR to Left & Right Cameras

$$\mathbf{R}_{\text{Bouguet}\rightarrow\text{Left}} = \begin{pmatrix} -0.01356316099573 & 0.99975098568338 & -0.01772025081386 \\ 0.00026099474360 & -0.01771834065446 & -0.99984298381606 \\ -0.99990798203912 & -0.01356565625227 & -0.00002061304530 \end{pmatrix}$$

$$\mathbf{T}_{\text{Bouguet}\rightarrow\text{Left}} = \begin{pmatrix} 0.36290818771769 & -0.81604891467805 & -1.62854686949177 \end{pmatrix}$$

(units in meters)

$$R_{\text{Bouguet}\rightarrow\text{Right}} \begin{pmatrix} 0.00145197936667 & 0.99933445267465 & -0.03644919002356 \\ 0.00571965315642 & -0.03645693154009 & -0.99931885687725 \\ -0.99998258851026 & 0.00124251363615 & -0.00576878116893 \end{pmatrix}$$

$$\mathbf{T}_{\text{Bouguet}\rightarrow\text{Right}} = \begin{pmatrix} 0.03271813694169 & -0.80854854597228 & -1.62103557239767 \end{pmatrix}$$

(units in meters)

NO. OF
COPIES ORGANIZATION

* ADMINISTRATOR
DEFENSE TECHNICAL INFO CTR
ATTN DTIC OCA
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218
*pdf file only

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI IS R REC MGMT
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI OK TECH LIB
2800 POWDER MILL RD
ADELPHI MD 20783-1197

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL SE SE P GILLESPIE
N NASRABADI
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 NATL INST OF STDS & TECHNOLOGY
ATTN DR M SHNEIER
BLDG 200
GAITHERSBURG MD 20899

2 UNIV OF MARYLAND
INST FOR ADV COMPUTER STUDIES
ATTN DR L DAVIS D DEMENTHON
COLLEGE PARK MD 20742-3251

ABERDEEN PROVING GROUND

1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI OK (TECH LIB)
BLDG 4600

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL WM J SMITH
T ROSENBERGER
BLDG 4600

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL WM B A HORST
W CIEPIELLA
BLDG 4600

NO. OF
COPIES ORGANIZATION

16 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL WM BF H EDGE
M BARANOSKI P FAZIO
M FIELDS G HAAS (4 CYS)
T HAUG W OBERLE (4 CYS)
R PEARSON R VON WAHLDE
S WILKERSON
BLDG 390

2 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL WM RP C SHOEMAKER
J BORNSTEIN
BLDG 1121